PPU College Of Engineering and Technology

The Home of Competent Engineers and Researchers

Electrical Engineering Department

Communication Engineering Program

Graduation Project

Intelligent Traffic Light System For Emergency Vehicle

Project Team

Tasneem GH. Manasrah **Nadia A. Rasras** **Nagham A. Shweiki**

Project Supervisor

Eng. Ayman Wazwaz

Hebron Palestine

2013

شهادة تقييم البحث

جامعة بوليتكنك فلسطين

الخليل_فلسطين

Intelligent traffic light system for emergency vehicle

اعداد :

نغم عبد الجبار شويكي

ناديا عبد الرحمن رصرص

تسنيم غاندي مناصرة

اشراف :

م٠ أيمن وزوز

بناء على توجيهات الاستاذ المشرف م. ايمن وزوز وبموافقة اعضاء اللجنة الممتحنة تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية في كلية الهندسة والتكنولوجيا للوفاء بجزء من متطلبات الحصول على درجة البكالوريوس في هندسة الاتصالات والالكترونيات .

توقيع المشرف .... توقيع رئيس الدائرة ....

٢٠١٣

Abstract

In this project, we introduce an intelligent system that controls the traffic lights, to help emergency vehicles such as ambulance, fire engine, police cars, etc… to reach the place they need in due course and without any late, so drivers that have emergency case will not spend unnecessary time waiting for the traffic lights to change, because the system will give them the priority to have the green light.

ZigBee technology will be used as wireless communication, a transmitter on the emergency vehicle side and a receiver on the traffic light side.

If there is an emergency case, the system will be triggered and XBee receivers will be ready to receive signal from emergency car transmitter.

Then signal will be sent to the central XBee receiver which will pass it to the traffic light's microcontroller that will change the normal behavior of the traffic lights by setting the desired one as green and the others as red.

Abstract

In this project, we introduce an intelligent system that controls the traffic lights, to help emergency vehicles such as ambulance, fire engine, police cars, etc… to reach the place they need in due course and without any late, so drivers that have emergency case will not spend unnecessary time waiting for the traffic lights to change, because the system will give them the priority to have the green light.

ZigBee technology will be used as wireless communication, a transmitter on the emergency vehicle side and a receiver on the traffic light side.

If there is an emergency case, the system will be triggered and XBee receivers will be ready to receive signal from emergency car transmitter.

Then signal will be sent to the central XBee receiver which will pass it to the traffic light's microcontroller that will change the normal behavior of the traffic lights by setting the desired one as green and the others as red.

الملخص

يهدف هذا المشروع إلى بناء نظام إشارات مرور ذكيّ يعمل على التصرف بحنكة و ذكاء أمام الحالات الطارئة، حيث يساعد سيارات الطوارئ كسيارات الإسعاف و سيارات الإطفاء و سيارات الشرطة بأن تصل إلى وجهتها في أسرع وقت ممكن و دون أي تأخير.

هذا النظام يعمل على التخلص من الوقت الذي تقضيه سيارة الإسعاف و هي تنتظر إشارة المرور كي تتغير و تُفتح أمامها، هذا الوقت الذي مهما كان ضئيلا إلا أنه يساوي حياة بأكملها، و بالتالي سنقوم بإعطائها حق الأولوية في الضوء الأخضر.

هذا النظام يستخدم ما يُعرف بتقنية الـ ZigBee التي ستستخدم كوسيلة اتصال لاسلكية، سيكون هناك XBee unit تعمل كمرسل مُتواجِدة في سيارة الإسعاف و XBee units تعمل كمستقبل مُتواجِدة بالقرب من إشارات المرور.

في حال تواجد حالة طارئة سيقوم السائق بتشغيل النظام، مما يؤدي إلى إرسال الإشارة من الـ XBee unit المرسِلة إلى XBee units المستقبِلة التي بدورها ستقوم بإرسال هذه الإشارة إلى XBee unit المركزية المتصلة مع متحكم دقيق التي ستعمل على إرسال هذه الإشارة إلى هذا المتحكم ليقوم بتغيير النظام الاعتيادي لإشارات المرور حيث سيقوم بفتح إشارة المرور أمام سيارة الإسعاف و إغلاق بقية إشارات المرور.

DIDICATION

To the candles that burn to light our life and future,

Our parents who made it all possible to receive the stage...

To the flowers of the earth & the stars of the sky,

Our brothers & sisters...

To Deanship of Research & Graduate Studies for funding this project

To our supervisor who always encouraged and supported us

Eng.Ayman Wazwaz

To whom we love and cant forget ,

Our beloved (M & A), friends, and instructors...

To all souls we respect , martyrs..

To mother land Palestine ..

ACKNOWLEDMENT

We like to express our thanks and gratitude to Allah, the Most Beneficent, the Most Merciful who granted us the ability and willing to start our project. We pray to his greatness to inspire us the right path to his content and enable us to continue the work started in the project to benefits our country.

We could not forget our families, who stood by us, with their support, love and care for our whole lives, they were with us with their bodies and souls, believed in us and helped us to accomplish this project.

We wish to express our deep and sincere thanks and gratitude to Palestine Polytechnic University (PPU), the Department of Electrical and Computer Engineering, College of Engineering and Technology . We wish to express our thanks to Eng. Ayman Wazwaz for evaluable help, encouragement, supervision and guidance in solving the problems that we faced from time to time during this project.

Work team

List of Figures

List of tables

Contents:

Chapter One:

Chapter Two:

Contents:

Chapter One:

Chapter Two:

Chapter Three:

Chapter Four:

Chapter Five:

Chapter Six:

Chapter One

Introduction

1.1 Overview

In this chapter, we will provide general literature information that is important to address our project. This includes an explanation of the main idea, motivations, and the project objectives. After that we will give a look at some related works.

1.2 Project Motivations:

Souls of our lovers are the most important thing of our life, nothing can compensate their absence, also regret will never change anything and it will be too late if we didn't do something for them.

So we thought to find a solution of the traffic jams on the traffic light; because it is not acceptable to the emergency vehicles, by allowing the emergency vehicle to reach the place it need in due course without problems in heavy traffic.

1.3 Main idea:

We intend to build a system that allows the emergency vehicle to communicate with the traffic light using ZigBee technology then open the road in front of it by controlling the traffic lights using PIC microcontroller.

1.4 Objectives:

We aspire in our project to achieve several objectives:

1. We would like to create a safety traffic environment.
2. Control the traffic light system in a way that serve our needs
3. Create a traffic environment that appreciate special cases and help them to reach the place they need in due course.

1.5 Idea and approach:

The project idea is standing on finding a way that allows emergency vehicles to communicate with the traffic light in order to open the road in fort of them without any hindrance or late , and this idea is going to work as what we are going to show :

We are going to provide the system a technique that consists of a transmitter on the emergency vehicle and a receiver on the traffic light, we have many choices about what is this kind of technique that we will use? It could be a ZigBee or Wi-Fi or IR, and whatever was the technique we are going to use, the communication mechanism will be done in this way: if there is an emergency vehicle with an emergency case the driver will run the transmitter device, and the receiver device on the traffic light will receive signal so this traffic will set as green, and the others will set as red.

But there are many things should be considered like the distance between the emergency car and the traffic light; if the distance is large we will need more nodes in the road, and we should know the time delay that the traffic light needs to set green from red, and finally a microcontroller is used to connect the traffic lights together and control them. We may use assembly language or C language to program the microcontroller.

1.6 Block Diagram:

Here is the main block diagram of the project as shown in figure 1.1.It includes all components we are going to use in this project. The emergency vehicle is communicating with the traffic light through ZigBee network, and the PIC microcontroller controls the system.

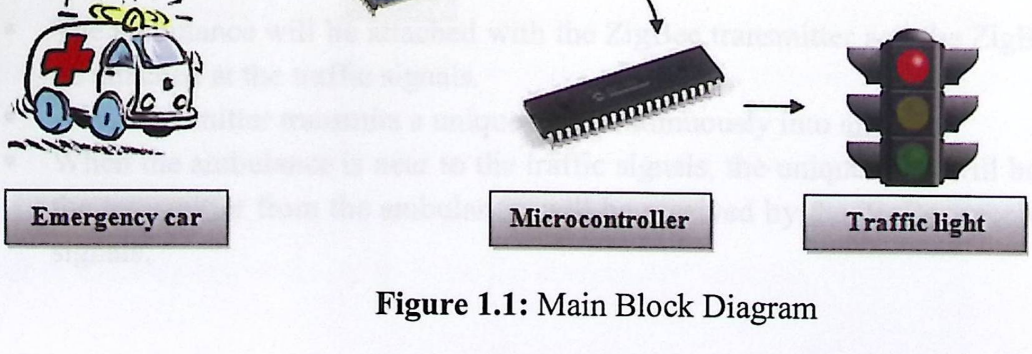


Figure 1.1: Main Block Diagram

1.7 Requirements:

1.7.1 Hardware:

1. ZigBee devices.
2. Microcontrollers.
3. Traffic light.

1.7.2 Software:

Programming software by using Micro C to program the microcontroller which responsible for many tasks.

1.8 Challenges:

We expect to face many challenges in this project. We can summarize them below:

1. Availability of electronic chips.
2. Dealing with a new technology (ZigBee).
3. Knowing the direction that the vehicle will come from.
4. Getting accurate measurements of power.
5. How to program microcontroller to achieve the goals on hardware.

1.9 Related works:

1.9.1 Innovative congestion control system for ambulance using ZigBee [1]

This project is designed to clear the traffic and provide a way to the ambulance so that the patient can be saved out of danger, and will reach the hospital safely and as quickly as possible.

To implement this application the project uses ZigBee technology, and here we will see how the system works:

- The ambulance will be attached with the ZigBee transmitter and the ZigBee receiver will be attached at the traffic signals.
- This transmitter transmits a unique code continuously into air.
- When the ambulance is near to the traffic signals, the unique code will be transmitted by the transmitter from the ambulance, will be received by the ZigBee receiver at the traffic signals.

The system uses a compact circuitry built around Flash version of AT89S52 microcontroller with a non-volatile memory capable of retaining the password data for over ten years. Programs are developed in embedded C. ISP is used to dump the code into the microcontroller.

1.9.2 Traffic controlling system for ambulance [2]

This project is designed to clear the road for the ambulance before it has reached the traffic signal spot, the ambulance/fire engine mobility at any place is unpredictable, and at the same time the traffic signal at the crowded palace is also the important one. So this project has designed in order to meet up the above condition without any loss.

The components used in this project are PIC microcontroller, RF transmitter and receiver and traffic signal lamps.

The operation of this project is the direction of the movement of the ambulance is given from the keypad to the microcontroller from which the signal is given to the low power transmitter which transmits the signal to the receiver present in the predestine stamp.

Then the received signal is given to the microcontroller which gives the signal to the relay, as soon as the relay receives the signal it will change the signal in order to give way to the arriving ambulance at the same time the remaining pathway will gets blocked.

After that the normal condition will be achieved for the passers.

1.9.3 Intelligent traffic light control for ambulance [3]

The idea behind this project is that drivers will not spend unnecessary time waiting for the traffic lights to change. The system developed is able to sense the presence or absence of vehicles within certain range by setting the appropriate duration for the traffic signals to react accordingly. The system can help to solve the problem of traffic congestion.

The main aim in designing and developing of the Intelligent Traffic Signal system it consists of a computer that controls the selection and timing of traffic movements in accordance to the varying demands of traffic signal as registered to the controller unit by sensors (IR).

The second part is the signal visualization or in simple words is signal face. Signal faces comprise of solid red, yellow, and green lights. The third part is the detector or sensor. The sensor or detector is a device to indicate the presence of vehicles.

The system uses a compact circuitry built around Flash version of AT89S52 microcontroller with a non-volatile memory capable of retaining the password data for over ten years. Programs are developed in embedded C. ISP is used to dump the code into the microcontroller.

1.9.2 Traffic controlling system for ambulance [2]

This project is designed to clear the road for the ambulance before it has reached the traffic signal spot, the ambulance/fire engine mobility at any place is unpredictable, and at the same time the traffic signal at the crowded palace is also the important one. So this project has designed in order to meet up the above condition without any loss.

The components used in this project are PIC microcontroller, RF transmitter and receiver and traffic signal lamps.

The operation of this project is the direction of the movement of the ambulance is given from the keypad to the microcontroller from which the signal is given to the low power transmitter which transmits the signal to the receiver present in the predestine stamp.

Then the received signal is given to the microcontroller which gives the signal to the relay, as soon as the relay receives the signal it will change the signal in order to give way to the arriving ambulance at the same time the remaining pathway will gets blocked.

After that the normal condition will be achieved for the passers.

1.9.3 Intelligent traffic light control for ambulance [3]

The idea behind this project is that drivers will not spend unnecessary time waiting for the traffic lights to change. The system developed is able to sense the presence or absence of vehicles within certain range by setting the appropriate duration for the traffic signals to react accordingly. The system can help to solve the problem of traffic congestion.

The main aim in designing and developing of the Intelligent Traffic Signal system it consists of a computer that controls the selection and timing of traffic movements in accordance to the varying demands of traffic signal as registered to the controller unit by sensors (IR).

The second part is the signal visualization or in simple words is signal face. Signal faces comprise of solid red, yellow, and green lights. The third part is the detector or sensor. The sensor or detector is a device to indicate the presence of vehicles.

RF BASED Ambulance alert system which civilian drivers elect to stay off the road in which the 3 signals automatically falls red and green to ambulance by sending signal from ambulance to traffic light sensor system.

We have many differences between the previous projects and our project, we will mention them as the following:

We see in [1] and [4] Systems used old AT89C51 microcontroller that has very less internal memory and no in-built ADC, and [4] system used CAN BUS controller which leads to complicated design and cost of the system more because of CAN BUS controller. Also power requirement will be more in systems that used AT89C51 microcontroller, but in our project we will use low consuming power microcontroller.

But [2] and [3] systems used RF transmitter and receiver, while in our project we will use ZigBee technology, this technology has many features which will be discussed later. And in [3] IR sensors are used to detect the presence of the emergency car, while in our project we don't need that; because we depend on ZigBee signal power measurements.

And project "Intelligent Cross Road Traffic Management System" [5] provide integrated intelligent traffic light system using photoelectric sensors distributed on long range before and after traffic light on roads. The system has the ability to open a complete path for such emergency cases until reaching the target but this system does not operate well when more than one emergence Vehicle come on the signal from two sides. But this problem is solved in our project.

1.10 Project Plan:

The project contains the following stages:

Stage 1: Preparing the project
The idea of the project is selected. Then the required information will be collected. Discussion with supervisor, and dividing tasks between the members of the group.

Stage 2: Analysis overview
Here, a deep and complete study for all options of the project will be made.

Stage 3: Study of the principles

In this stage, we study the behavior of traffic lights, ZigBee, microcontroller, sensors and any other technologies or information needed.

Stage 4: Documentation and writing
Writing and preparing the documentation of the project was start from the first stage, and will continue till the end of the project.

Stage 5: Microcontroller programming
Writing the code of the normal behavior of the traffic light, and then downloading to microcontroller.

Stage 6: software implementation
The programming of Xbee isstarted, writing the project codes and then downloading to microcontroller.

Stage 7: hardware implementation
Here we build the project circuits and making some testing .

Stage 8: system measurement
Here many experiments were taken the values of received signal strength in Xbee to develop system with accurate results and avoid false alarms.

Stage 9: system testing
Here the system will be tested to conclude the system performance.

Phase 10: Writing documentation
The documentation will continue from the first phase till the end in parallel

Table 1.1: Summary of the Project Plan

week / task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Stage1														
Stage2														
Stage3														
Stage4														
Stage5														

week / task	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Stage6	■	■	■											
Stage7			■	■	■	■	■	■						
Stage8							■	■	■	■				
Stage9										■	■	■		
Stage10					■	■	■	■	■	■	■	■	■	■

1.11 Estimated cost and budget:

The whole estimated cost will be approximately 340 JD, and the table below shows the cost of each hardware component.

Table 1.2: Estimated cost and budget

1-Xbee PRO 63Mw	50X5 JD
2-PIC18f4550	10X1 JD
3-PIC18f5025	15X4 JD
Total	340 JD

1.12 Report contents:

This project is mainly divided into six chapters, each of them describes specific part of the project as following:

Chapter One: includes the introduction, provides a general overview about the project, its objectives, importance, related works, challenges, time planning, estimated cost and at the end the report contents.

Chapter Two: discusses the theoretical background. It starts with general information about the project, wireless communication, ZigBee, sensor and then discusses the important aspects of the system including PIC microcontroller, sensors.

Chapter Three: presents the general system design concepts. It includes system objectives, general system block diagram, description of system design ' components and operation'.

Chapter Four:
discusses a detailed description about subsystem and overall hardware and software system design, with showing components pins and feat, and software requirements to perform the required function of the robot.

Chapter Five: It contains the result of Xbee power, testing to the whole system, and performance.

Chapter Six: This chapter will consider system achievement, real outcome, conclusion, and recommendation for developing the system in future.

2

Chapter Two

Theoretical Background

2.1 Wireless communications

2.2 ZigBee Theory

2.3 Micro-controller

2.4 Traffic light

2.5 Sensors

2.1 Wireless communications:

2.1.1 Introduction:

Wireless communications is a rapidly growing segment of the communications industry, with the potential to provide high-speed high-quality information exchange between portable devices located anywhere in the world. Potential applications enabled by this technology include multimedia Internet-enabled cell phones, smart homes and appliances, automated highway systems, video teleconferencing and distance learning, and autonomous sensor networks, to name just a few. However, supporting these applications using wireless techniques poses a significant technical challenge.

And in the time the wireless system has many advantages it also has disadvantages and here are some of them:

Advantages:

- Rapid installations
- Easy maintenance
- Avoided ground potential rise problems
- Low cost
- Mobility
- Safety due to remote operations
- Warmth of a van outside a freezing substation
- More data and additional capabilities
- Low costs – 7 times less in TVA's example

Disadvantages:

- Eavesdropping on data
- Unauthorized control commands
- Unreliable, disrupted communications in the noisy
- Substation environment
- Hackers, viruses, and worms

2.1.2 Types of Wireless Systems:

➢ Older wireless systems:

- Microwave systems
- Multiple address radio systems
- Satellite, particularly VSAT
- Spread spectrum radio, 928 MHz point-to-point

➢ New wireless systems:

- Wi-Fi – IEEE 802.11:

Wi-Fi is the name given by the Wi-Fi Alliance to the IEEE 802.11 suite of standards. 802.11 defined the initial standard for wireless local area networks (WLANs), but it was considered too slow for some applications and so was superseded by the extensions 802.11a and 802.11b, and later by 802.11g (with the release of 802.11n still pending).

The benefits of a Wi-Fi network: Extended Access; the absence of wires and cables extends access to places where wires and cables cannot go or where it is too expensive, Cost Reduction, Mobility.

- Bluetooth™ – IEEE 802.15.1:

The IEEE 802.15.1 standard is the basis for the Bluetooth wireless communication technology. Bluetooth is a low tier, ad hoc, terrestrial, wireless standard for short range communication. It is designed for small and low cost devices with low power consumption. [6]

- ZigBee – IEEE 802.15.4:

ZigBee is a low tier, ad hoc, terrestrial, wireless standard in some ways similar to Bluetooth.

The IEEE 802.15.4 standard is commonly known as ZigBee, but ZigBee has some features in addition to those of 802.15.4. It operates in the 868 MHz, 915 MHz and 2.4 GHz ISM bands. [7]

- WiMax – IEEE 802.16.

2.2 ZigBee Theory:

2.2.1 Introduction:

The past several years have shown a rapid development of wireless networking. However, up to now wireless networking has been mainly focused on high speed communications, and long range applications, such as the IEEE 802.11 Wireless Local Area Network (WLAN) standards. And there are many wireless monitoring and control applications in industrial and home environments, which require longer battery life, lower data rate and less complexity than the existing standards. So, the ZigBee Alliance and the IEEE decided to join forces, and ZigBee is the commercial name for this new technology

"ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802 standard for personal area networks. Applications include wireless light switches, electrical meters with in-home-displays, and other consumer and industrial equipment that require short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking." – Wikipedia

2.2.2 Name of ZigBee:

The name ZigBee is said to come from the domestic honeybee which uses a zigzag type of dance to communicate important information to other hive members. This communication dance (the ZigBee principle) is what engineers are trying to emulate with this protocol-a bunch of separate and simple organism that join together to tackle complex tasks.

2.2.3 Types of ZigBee devices:

1. Coordinator: coordinator organizes the network and maintains routing tables. Each network must be formed by a coordinator and never have more than one coordinator in the network regardless of the network topology.
2. Router: router can talk to the coordinator and to other routers. It reduces function end devices. It can join existing networks, send information, receive information, and route information. A network may have multiple router radios.
3. End devices: can join networks and send and receive information. They can talk to routers and the coordinator, but not to each other. End devices always need a router or the coordinator to be their parent device. ZigBee networks may have any number of end devices.

2.2.4 ZigBee networking:

ZigBee can use mesh networking, which may extend over a large area and contain thousands of nodes. Each FFD in the network also acts as a router to direct messages .the routing protocol optimizes the shortest and most reliable path through the network and can dynamically change, so as to take evolving conditions into account. This enables an extremely reliable network, since the network can heal itself if one node is disabled. ZigBee networks are primarily intended for low duty cycle sensor networks. anew network node may be recognized and associated in about 30 ms. Waking up a sleeping node takes about 15 ms , as does accessing a channel or transmitting data. ZigBee applications benefit from the ability to quickly attach information,detach, and go to deep sleep, which results in low power consumption and extended battery life.

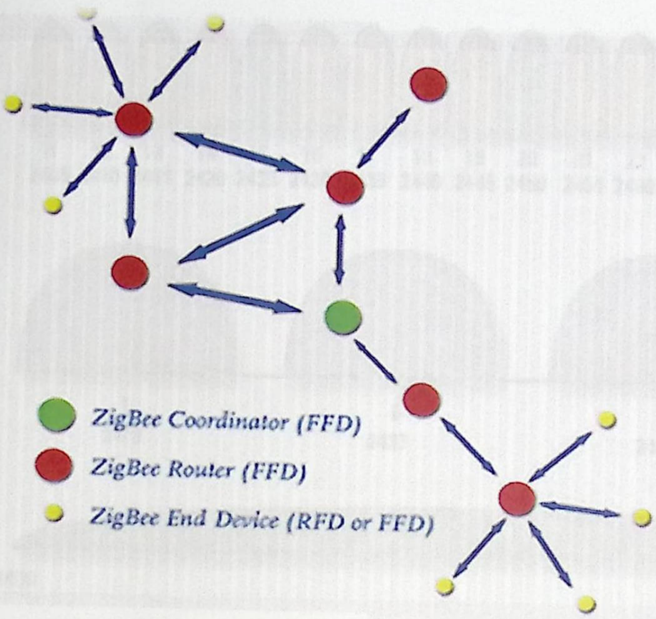


Figure 2.1: ZigBee network devices

2.2.5 ZigBee Channels and Frequencies

The RF spectrums and available channels for ZigBee (802.15.4) and Wi-Fi (802.11b/g) overlap. You can avoid interference by selecting ZigBee channels that use the free space between two neighboring 802.11 channels, plus channels 25 and 26.[8]

2.2.5.1 ZigBee, Wi-Fi & Bluetooth Channels

In figure below, 802.15.4 Orange channels have more substantial overlap with Wi-Fi channels 1, 6 & 11, while Grey channels have less overlap with Wi-Fi channels 1, 6 & 11.

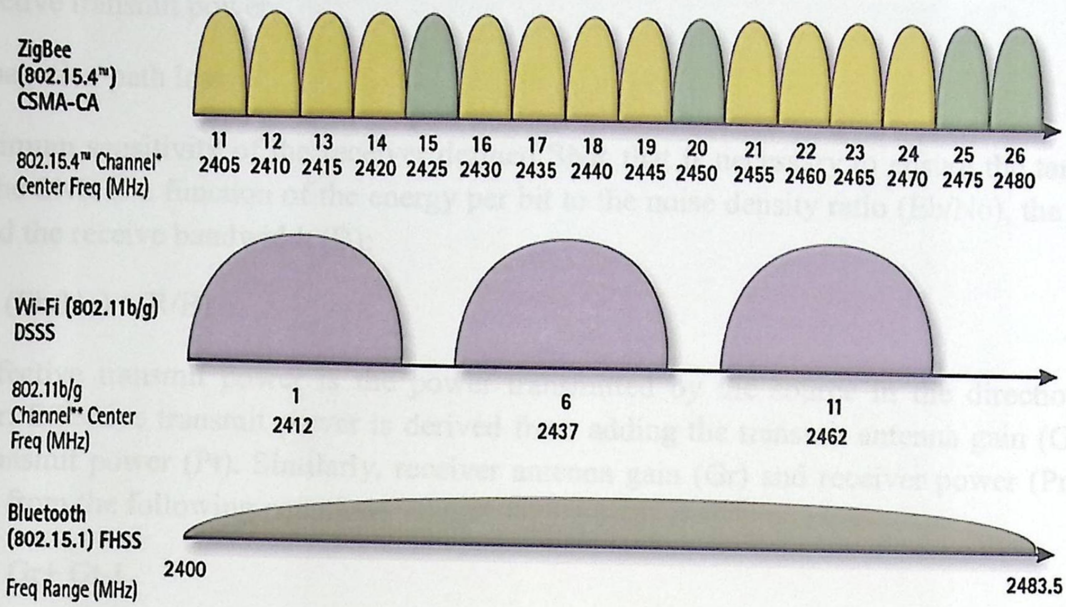


Figure 2.2: ZigBee, Wi-Fi & Bluetooth Channels

Channels 1, 6 & 11 are recommended for use in the USA. Only 2.4 GHz channels are shown above. Different countries have different 802.11 b/g channels.

2.2.6 Why we chose ZigBee:

In our view ZigBee is the most appropriate wireless communication technologies for our project. There are many reasons why that is so, some of them are:

- High Reliability: ZigBee is based on a solid global standard 'IEEE 802.15.4 standard' and it uses modulation technologies like DSSS and CSMA/CA which are among the best in the respective domains. So our project tends to be more reliable and easy to demonstrate anywhere.
- Easy to Use: Most ZigBee modules can directly be interfaced to microcontrollers, processors and even computers. And they available with a direct UART communication interface; this makes communication even easier...
- Easy Availability: These days many ZigBee modules are available easily and made by various manufacturers so it is easily accessible and it's not worthless to us to use it
- Infinite Possibilities: ZigBee is the most flexible communication technology and it can be used for any type of wireless communication like simple point to point communication, Star network, Mesh Network, Tree Networks because ZigBee protocol has flexible architecture.

And ZigBee has good data rate and low cost…

2.2.7 Comparing ZigBee System Ranges [9]

The range of the RF system is defined as the maximum distance between the signal source or transmitter and the receiver. Range depends on three factors:

1. Effective transmit power

2. Propagation path loss

3. Minimum sensitivity of the receiver defined SNR that is necessary to ensure the target error rate. The SNR is a function of the energy per bit to the noise density ratio (Eb/No), the data rate (R), and the receive bandwidth (B):

SNR = (Eb/No) ×(R/B)

The effective transmit power is the power transmitted by the source in the direction of the receiver. Effective transmit power is derived from adding the transmit antenna gain (Gt) to the total transmit power (Pt). Similarly, receiver antenna gain (Gr) and receiver power (Pr) can be derived from the following equation:

Pr= Pt+ Gr+ Gt-L

Where: Pr = Receiver power in dBm

Pt = Transmitter power in dBm

Gr = Receiver antenna gain in dB

Gt = Transmitter antenna gain in dB

L = Attenuation at 2450 MHz in dB

Assuming the antenna gain for both transmitter and receiver is0 dB,for free-space propagation condition receives power is given by:

Pr= Pt- (10nlog d +50.3); n: path loss exponent.

2.2.8 OSI Model:

It is a product of the Open Systems Interconnection effort at the International. It is a prescription of characterizing and standardizing the functions of a communications system in terms of abstraction layers. Similar communication functions are grouped into logical layers. A layer serves the layer above it and is served by the layer below it.

The OSI divides telecommunication into seven layers; the layers are in two groups. The upper four layers are used whenever a message passes from or to a user. But, the lower three layers are used when any message passes through the host computer.

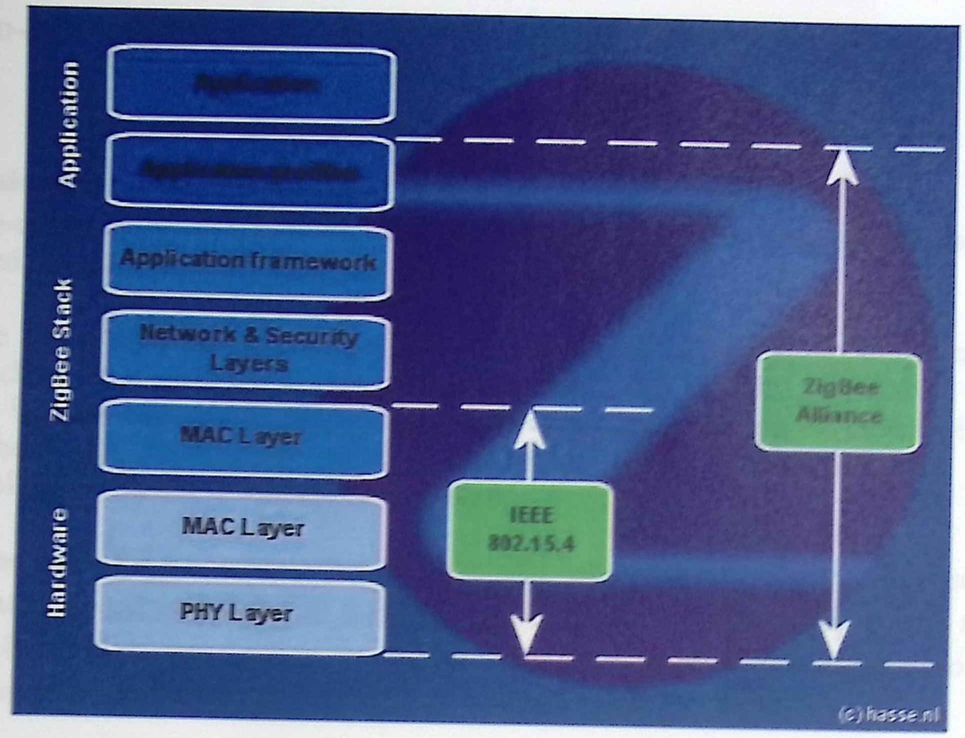


Figure 2.3:OSI versus ZigBee

2.2.9 Repeaters:

A repeater is an electronic device that receives a signal and retransmits it at a higher level or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances.

Repeaters advantages:

- Easy to expand a network over a large distance.
- Connection between various types of media [e.g. fiber optic, UTF, coaxial cable] is possible.

Repeaters disadvantages:

- Traffic cannot be filtered to ease congestion.
- A repeater cannot work across multiple network architectures.

And here in our project we are going to use the repeater as unicast communication (point -to-point) connection, which is between a single sender and a single receiver.

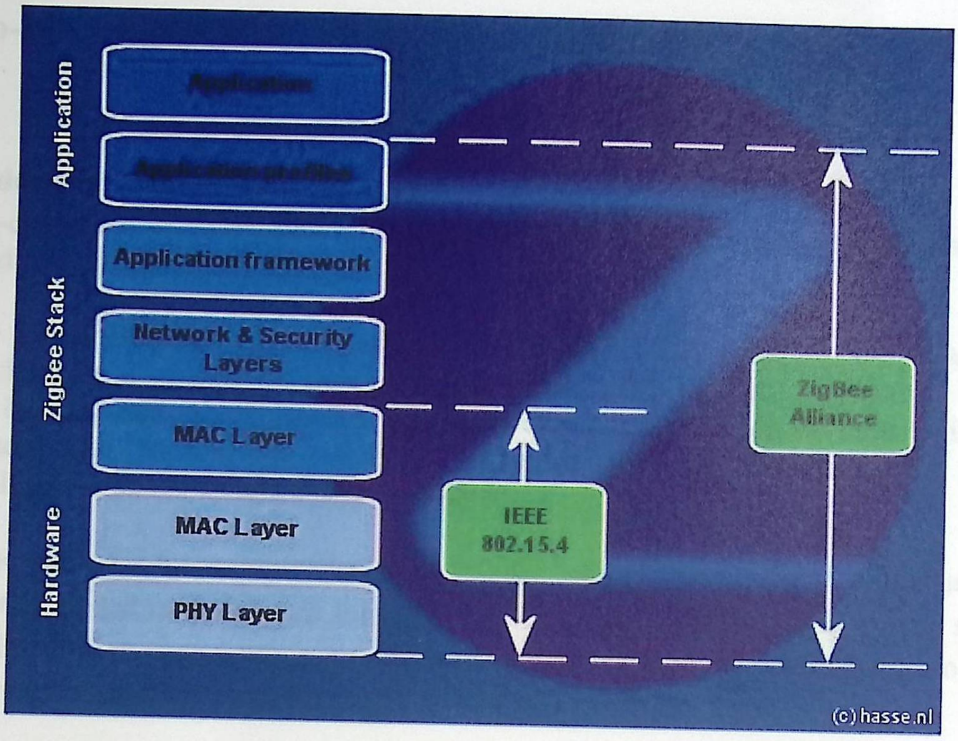


Figure 2.3:OSI versus ZigBee

2.2.9 Repeaters:

A repeater is an electronic device that receives a signal and retransmits it at a higher level or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances.

Repeaters advantages:

- Easy to expand a network over a large distance.
- Connection between various types of media [e.g. fiber optic, UTF, coaxial cable] is possible.

Repeaters disadvantages:

- Traffic cannot be filtered to ease congestion.
- A repeater cannot work across multiple network architectures.

And here in our project we are going to use the repeater as unicast communication (point –to-point) connection, which is between a single sender and a single receiver.

2.3 Micro-controllers:

2.3.1 Introduction:

A micro-controller is a very powerful device, which is capable of executing a series of pre-programmed tasks and interacting with other hardware devices.

A single microcontroller can be sufficient to control a small mobile robot, an automatic washer machine or a security system. Any microcontroller contains a memory to store the program to be executed, and a number of input/output lines that can be used to interact with other devices, like reading the state of a sensor or controlling a motor. Nowadays, microcontrollers are so cheap and easily available.

The prime use of a microcontroller is to control the operation of a machine using a fixed program that is stored in ROM and that does not change over the lifetime of the system. The microcontroller design uses a much more limited set of instructions that are used to move code and data from internal memory to the ALU.

Many instructions are coupled with pins on the IC package. The pins are programmable independently, that is capable of having several different functions depending on the program. The microcontroller is concerned with getting data from and to its own pins; architecture and instruction set are optimized to handle data in bit, byte and word size.

Every application demands a microcontroller, today there is no such electronic instrument or robot that functions without microcontroller. Generally for any application, often designers chose the 8 – bit controller, because they are most popular microcontrollers in use today, another important aspect is cost effective.

Programming PIC microcontrollers is a simple three steps process, write the code, compile the code, and upload the code into a microcontroller. Writing the code can be developed in many Integrated Development Environments (IDE's) for example, MPLAP IDE, which is software, developed for the Microchip appliances like the PIC microcontrollers.

Compiling the code can be done by the compiler of the MPLAP IDE. There are different compilers associated to work with PIC chips, C compiler, or assembler for assembly language codes, and many more. The decision of which compiler to use, is a developer choice, depending on the application which the PIC is a part of. The final step of programming the PIC chip is uploading the code into the microcontroller. This can be done also in MPLAP IDE or in different programs that are connected to the PIC kit (figure 2.4). The uploading process can be done through a USB cable or other connecting technique depending on the kit that contains the PIC chip.

Figure (2.4) PICDEM™ FS-USB Evaluation Kit for PIC18F4550

In this project the PIC18F4550 is used. This is due to its availability, cheap cost, easy programming (75 Instructions single-word instruction. The PIC18F4550 has all what is needed for the implementation of this project, enough I/O Ports, A\D, and USART.

2.3.2 Description of PIC18 Microcontrollers:

The PIC18F4550/2455/2550/4455 family offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high-endurance, Enhanced Flash program memory.

On top of these features, this family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

Devices in the PIC118F4550/2455/2550/4455 family are available in 40/44-pin packages. All of the devices in the PIC118F4550/2455/2550/4455 family incorporate a range of features that can significantly reduce power consumption during operation (Nano Watt TECHNOLOGY). Key items include:

• Alternate Run Modes: By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.

• Multiple Idle Modes: The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4%, of normal operation requirements.

• On-the-Fly Mode Switching: The power managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.

• Low Consumption in Key Modules: The power requirements for both Timer1 and the Watchdog Timer are minimized.

2.3.3 Special Features:

• Memory Endurance: The Enhanced Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 100,000 for program memory and 1,000,000 for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.

• Self-Programmability: These devices can write to their own program memory spaces under internal software control. By using a boot loader routine located in the protected Boot Block at the top of program memory, it becomes possible to create an application that can update itself in the field.

•10-Bit A/D Converter: This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reducing code overhead.

• Extended Watchdog Timer (WDT): This enhanced version incorporates a 16-bit prescale, allowing an extended time-out range that is stable across operating voltage and temperature

2.4 Traffic lights:

2.4.1 Overview:

Traffic lights or what can be called stoplights, traffic lamps, traffic signals, signal lights, robots [10] was first installed in London in 1886 by the railway engineer J. P. Knight , They resembled railway signals of the time, with semaphore arms and red and green gas lamps for night use.

The modern electric traffic light was by police officer William L.Potts of Detroit, Michigan, decided to do something about the problem caused by the ever increasing number of automobiles on the street. what had in mind was figuring out away to adapt railroad signals for street use. Potts used red, amber, and green railroad and about thirty –seven dollars worth of wire and electrical controls to make the words first 4-way three color traffic light. it was installed in 1920 on the corner of Woodward and Michigan Avenues [11].

A traffic signal, controls vehicle traffic passing through the intersection of two or more roadways by giving a visual indication to drivers when to proceed, when to slow, and when to stop. In some cases, traffic signals also indicate to drivers when they may make a turn. These signals may be operated manually or by a simple timer which allows traffic to flow on one roadway for a fixed period of time, and then on the other road-way for another fixed period of time before repeating the cycle. Other signals may be operated by sophisticated electronic controllers that sense the time of day and flow of traffic to continually adjust the sequence of operation of the signals. Traffic engineers use signals to avoid traffic congestion and improve safety for both motorists and pedestrians alike.

The typical sequence of color phases:

- Illumination of the green light allows traffic to proceed in the direction denoted, if it is safe to do so
- Illumination of the orange/amber light denoting prepare to stop short of the intersection , if it is safe to do so
- Illumination of the red signal prohibits any traffic from proceeding.

2.4.2 Basic concepts for design [12]

The signal design procedure involves five major steps. They include:

1. Phase design :
   The objective of phase design is to separate the conflicting movements in an intersection into various phases, so that movements in a phase should have no conflicts.
2. Interval design for amber time:
   It called the change, this interval or amber time is provided after green time for movement. The purpose is to warn a driver approaching the intersection during the end of a green time about the coming of a red signal. It's normally about 4 to 5 seconds.
3. Cycle time :
   Cycle time is the amount of time from the beginning of a red light to the beginning of the next red light. All of the events at a signal happen inside one cycle. Normal cycle times are somewhere between 80 seconds and 120 seconds for most large roads...
4. Apportioning of green time
   Is the available time for the vehicle to cross the intersection and it's about 10-15 seconds.

5 - pedestrian crossing requirements.

2.4.3 Traffic light circuit design:

Here is an explanation for a simple traffic light controller circuit:

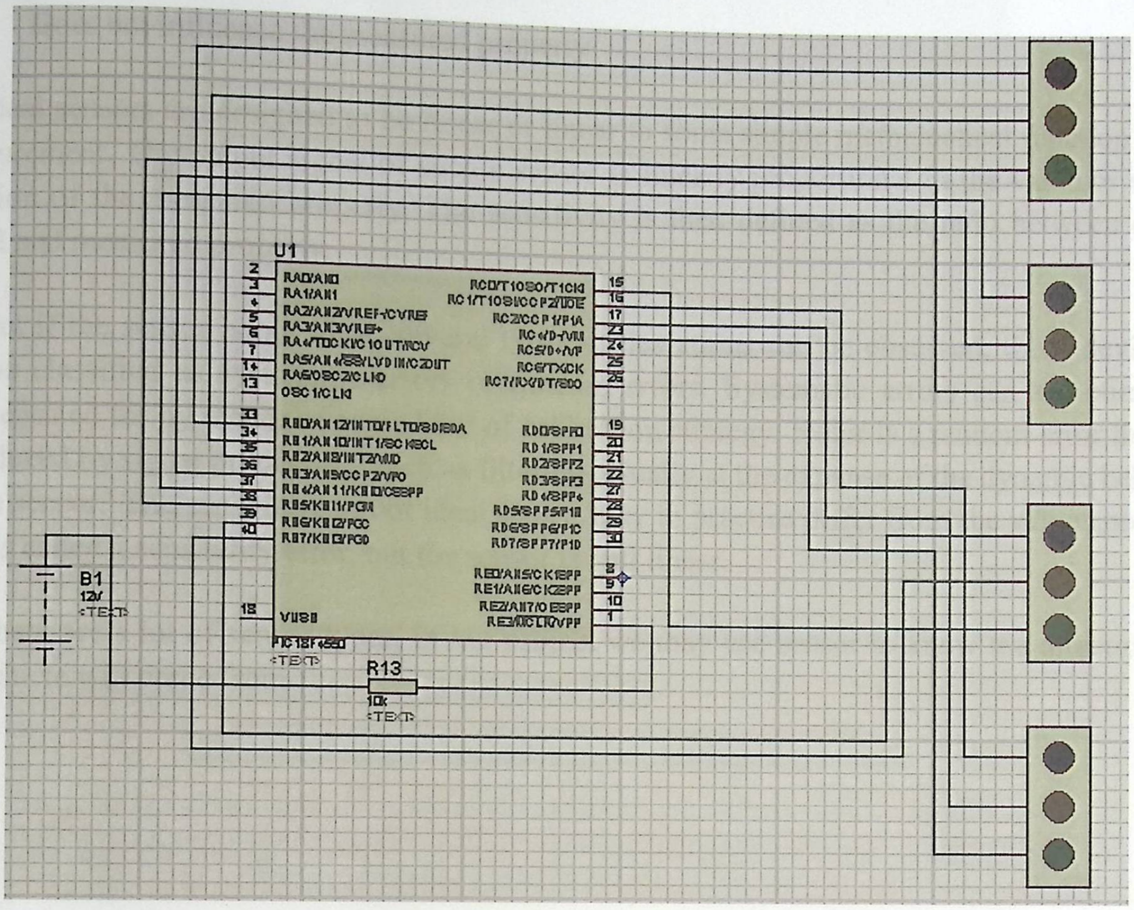


Figure 2.7 traffic light circuit

This figure will be discussed later in chapter 4.

2.5 sensor:

2.5.1 sensor identification:

A sensor is a converter that measures a physical quantity and converts it into a signal which can be read by an observer or by an electronic instrument. It is a device which receives and responds to a signal when touched. A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes, Sensors that measure very small changes must have very high sensitivities.

There are also innumerable applications for sensors; applications include cars, machines, aerospace, medicine, manufacturing and robotics.

A good sensor obeys the following rules:

- Is sensitive to the measured property only
- Is insensitive to any other property likely to be encountered in its application

- Does not influence the measured property

Ideal sensors are designed to be linear or linear to some simple mathematical function of the measurement. The output signal of such a sensor is linearly proportional to the value or simple function of the measured property. The sensitivity is then defined as the ratio between output signal and measured property.

But if the sensor is not ideal, several types of deviations can be observed; these deviations can be classified as systematic errors or random errors. Systematic errors can sometimes be compensated for by means of some kind of calibration strategy. Noise is a random error that can be reduced by signal processing, such as filtering, usually at the expense of the dynamic behavior of the sensor.' The sensitivity in not ideal sensor may in practice differ from the value specified. This is called a sensitivity error, but the sensor is still linear.

There are a lot of sensor types, in our project we may use sensor to know the direction that the vehicle will come from.

3

Chapter three

Conceptual Design

3.1 Introduction:

In this chapter, we will describe the general block diagram of the whole system, including the system elements; also we will explain the work design methodology.

3.2 Basic operation:

Once there is an emergency action, the driver of the emergency vehicle will trigger the system, the XBee device will be used at emergency vehicle and traffic light sides.

In the emergency vehicle side the XBee transmits data to the XBee on to the traffic light side through node to node ZigBee network to pass data to the PIC microcontroller that will make a decision which traffic light should be set as green.

3.3 Block diagram:

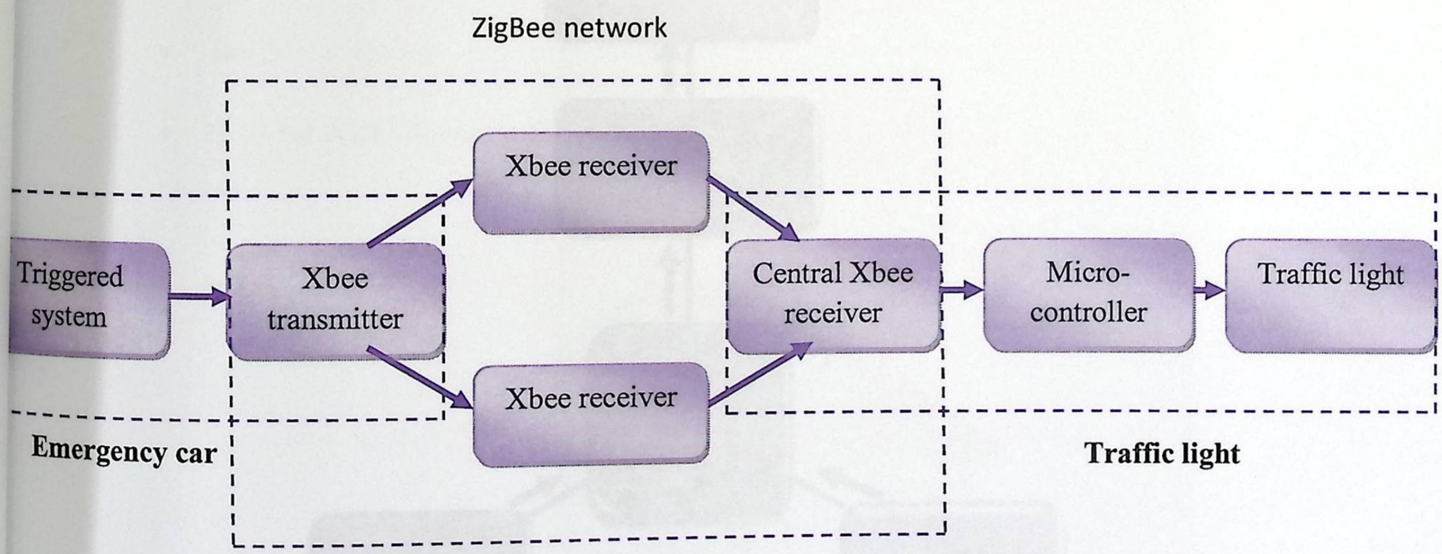


Figure 3.1: Main Block Diagram

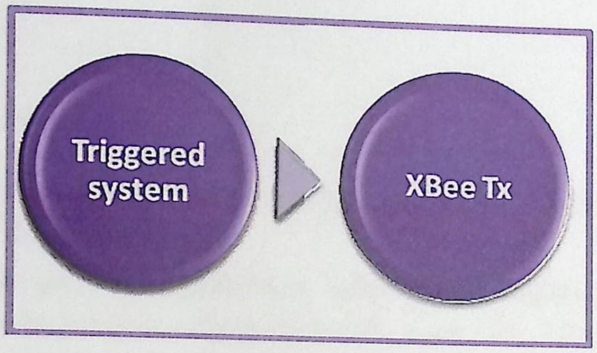


Figure 3.2: Emergency Vehicle Side

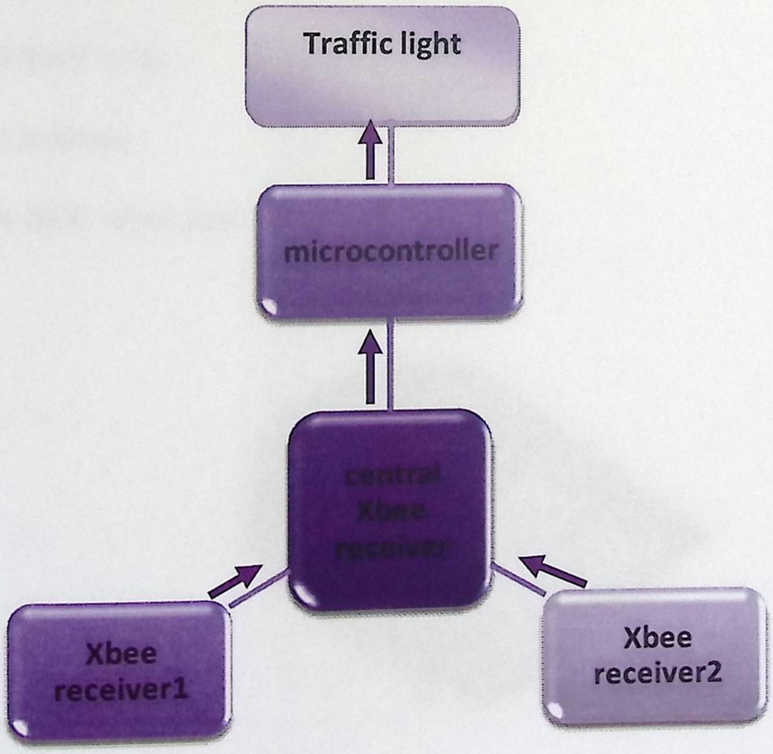


Figure 3.3: Traffic Light Side

3.4 Main system components:

In this section we are going to describe the function of each component we will use in our project, as the following:

3.4.1 ZigBee:

We are going to use:

1. XBee transmitter device on the emergency vehicle; which sends a signal to the traffic light side by ZigBee network.
2. XBee router to pass the signal from XBee transmitter to the central XBee receiver.
3. Central XBee receiver device at the traffic light side; which receives the signal from XBee routers with its addresses, and passes this signal to the traffic light's microcontroller.

In this project XBee 1mW Chip Antenna will be used, and its characteristics are:

- Allow a very reliable and simple communication between microcontrollers, computers, systems, and anything with a serial port.
- Point to point and multi-point networks are supported.
- 300ft (100m) range.
- Built-in antenna.
- 6 10-bit ADC input pins

Figure 3.4:XBee module

3.4.2 Microcontroller:

PIC microcontroller will be used to control the overall system, also execute the interrupt programwhenever there is an emergency car arriving.

In this project PIC18F4550 will be used, this is due to:

- Economical price.
- Enhanced Flash program memory.
- All of the devices in the PIC18F4550 reduce power consumption during operation.
- The module supports both low-speed and full-speed communication for all supported data transfer types.
- It also incorporates its own on-chip transceiver and 3.3V regulator and supports the use of external transceivers and voltage regulators.
- 10-Bit A/D Converter.
- Enhanced Addressable USART: This serial communication module is capable of standard RS-232 operation.

The PIC18F4550 has all what is needed for the implementation of this project, enough I/O Ports, ADC, timers, and serial communication USART.

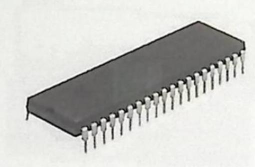


Figure3.5:PIC18F4550

3.4. 3 Traffic light:

The traffic light is the purposed component to be controlled in our project idea.

We will build it using the PIC microcontroller, so it will be easier to deal with and control.

The circuit consists of the PIC microcontroller that it will be programmed to work in the way we need and with the time delay to deal with the whole system design.

The time lengths are divides as the following:

1. The amber interval is normally about 2 seconds.
2. The green light interval is about 5 seconds.
3. The cycle time from the beginning of a red light to the beginning of the next red light for a four intersection road is about 40 seconds.

Back to the traffic light circuit we also need three leds; red, amber and green for each road in the intersection, and we need resistors, capacitors and wires for connections.

3.6 Software design:

A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows. This diagrammatic representation can give a step-by-step solution to a given problem. Process operations are represented in these boxes, and arrows connecting them represent flow of control.

And there are thethree main flowcharts that will be explained:

- Emergency vehicle as a sender (coordinator) flowchart.
- Routers that set on the street sides flowchart.
- Traffic light as a receiver flowchart.

- **The coordinator flowchart:**

As shown below in figure 3.6, when the driver triggers the system, the system will be ready to work. A signal that generated from the PIC microcontroller will be passed to the XBee unit transmitter to be transmitted to the XBee routers.

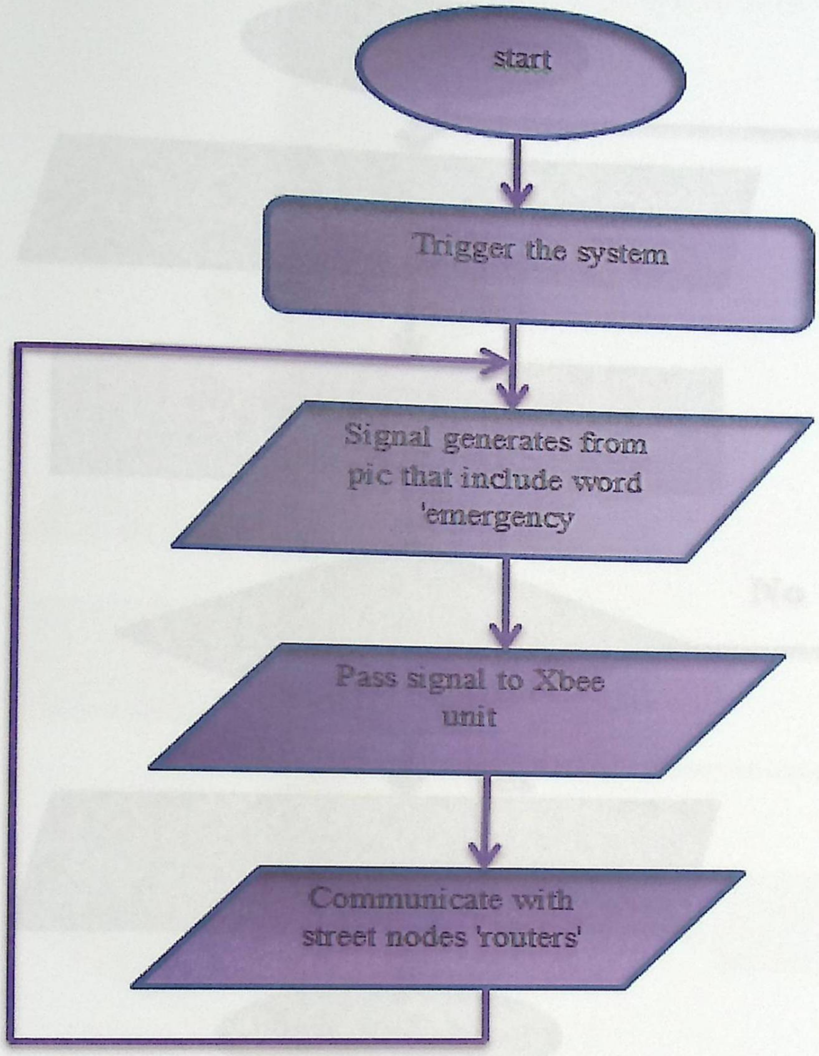


Figure 3.6: Flowchart For The Sender Side

This scenario consists of a cross road with three intersections, each section has its XBee receiver 'as router' ,all XBee routers are connected to the central XBee, which connected to traffic light PIC microcontroller.

- **The router flowchart:**

As shown in figure 3.7,after the received signal reached the Xbeenodes on the street we will measure the power value on each Xbee repeater and compare it with threshold value. If it is greater than the threshold, the signal included Xbee address should be passed to the central Xbee, unless the signal should be received again from Xbee coordinator.

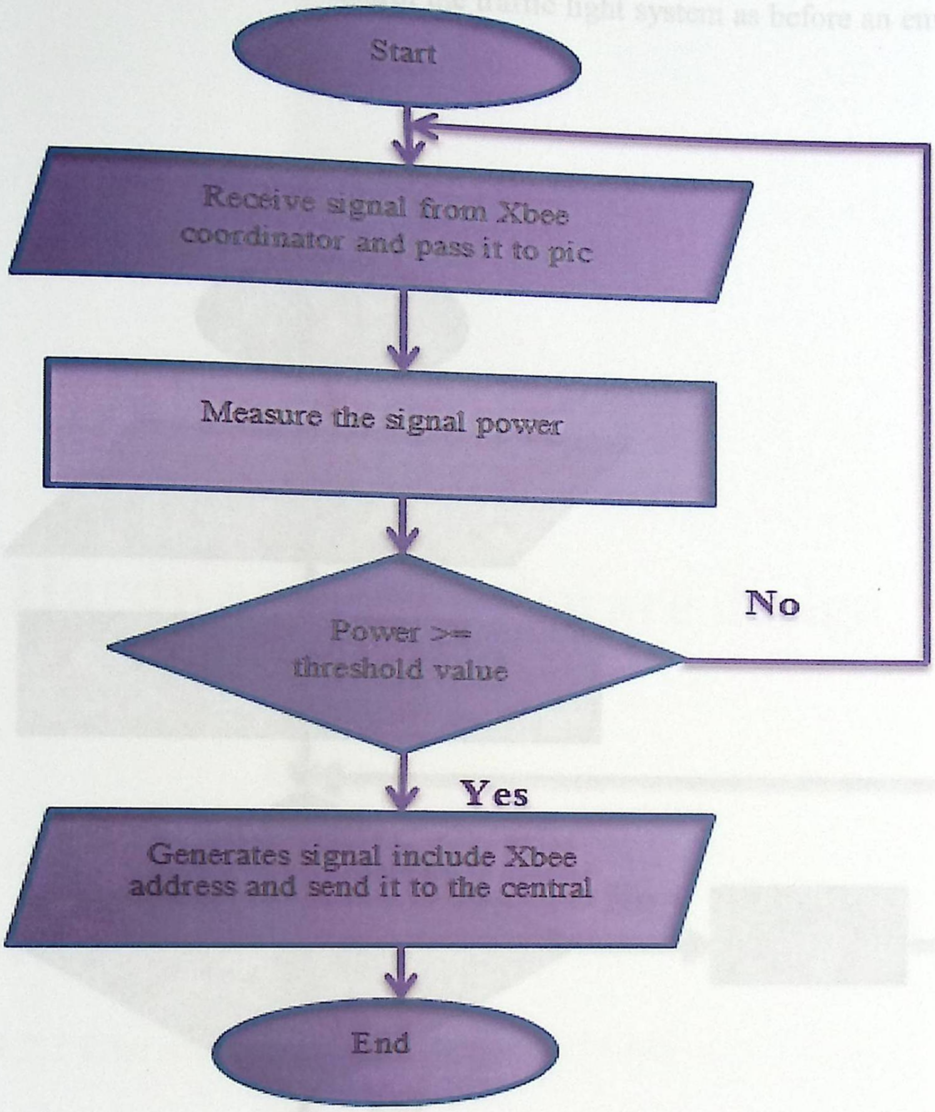


Figure 3.7: Router Flow Chart

- **Central flow chart :**

As shown in figure 3.8, after the received signal reached the central XBee receiver, we will take the mac address of the XBee repeater that has sent it so we can determine from which intersection the emergency car will come from.

After that the signal should be passed to the microcontroller that will start to execute the interrupt action which sets the desired traffic light as green and the others as red.

To make sure that the emergency car cross the traffic light, we will measure the signal power, compare it with threshold value that would be determined, so if the received power greater than this threshold value, keep the traffic light as green, if not the interrupt action will be finished, and the microcontroller will control the traffic light system as before an emergency car came.

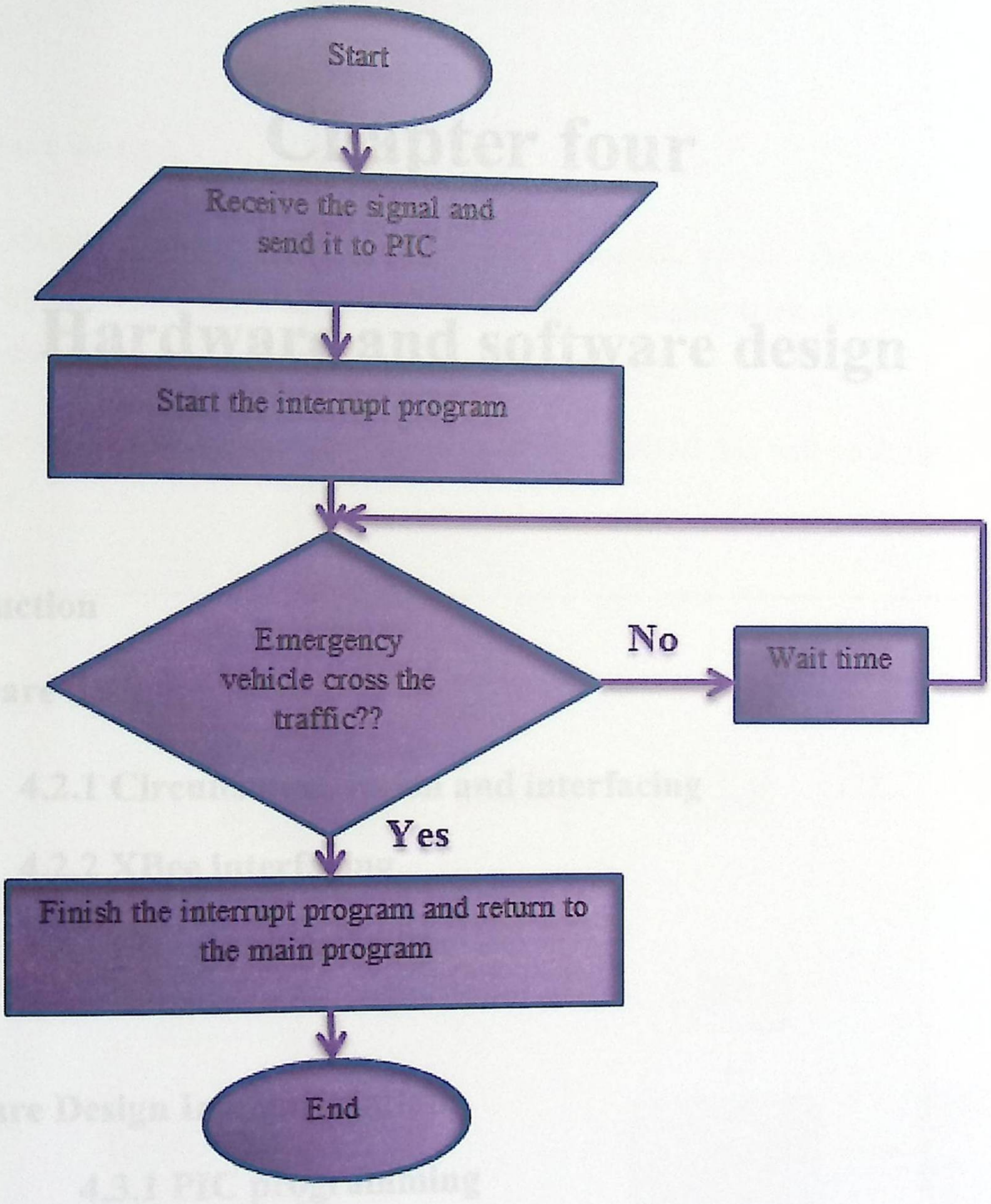


Figure 3.8: Flowchart For The Receiver Side

4

Chapter four

Hardware and software design

4.1 Introduction

In this chapter the construction and testing process for the system will be shown, the construction and testing process are very important to insure that the system works successfully. After collecting all the necessary information related to the project and analyzing them, the group started to build the system step by step, as will be shown.

4.2 Hardware design

This chapter provides a description about detailed project design which can be summarized by understanding the way XBee & microcontroller work and implementing the connection between the microcontroller, and radio transceiver.

Figure (4.1) shows the general schematic of every circuit that will be designed in details in this chapter.

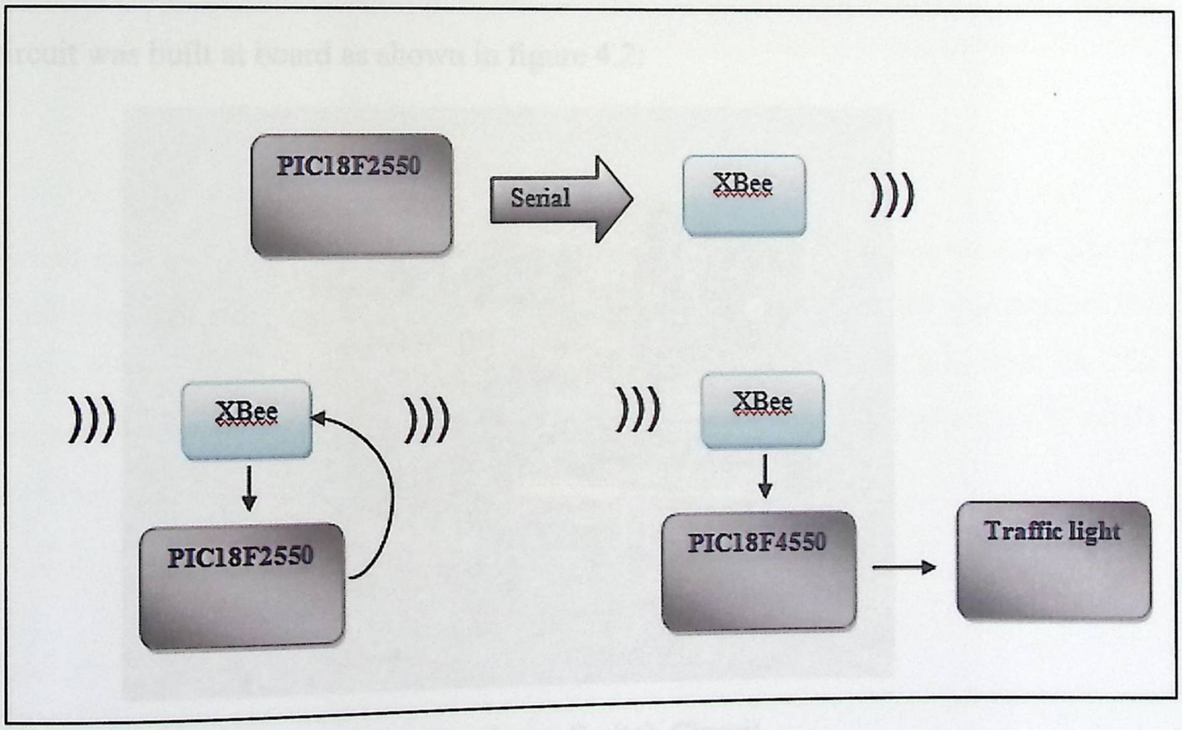


Figure 4.1: Hardware Schematic

1) The XBee on the ambulance broadcasts the signal to all possible destinations.

2) Signal will be received by the XBee that exists on the road.

3) The XBee will pass this signal to the PIC18F2550 which will arrange it in frame and put its address in it and check it if the power exceeded the threshold value that previously known, the microcontroller will re-pass it to the XBee to send it to the central XBee.

4) If the power isn't exceeded the threshold value, the microcontroller will not pass it to XBee so the XBee will not transmit any signal to the central XBee.

5) After the signal reaches the central XBee, it will transmit signal to the PIC18F4550 which controls the Traffic light.

4.2.1 Circuits description and interfacing

In this section the hardware circuits will be described in details, by showing the connections and interfacing between each XBee and PIC microcontroller.

- **Coordinator circuit (switch):**

The circuit was built at board as shown in figure 4.2:

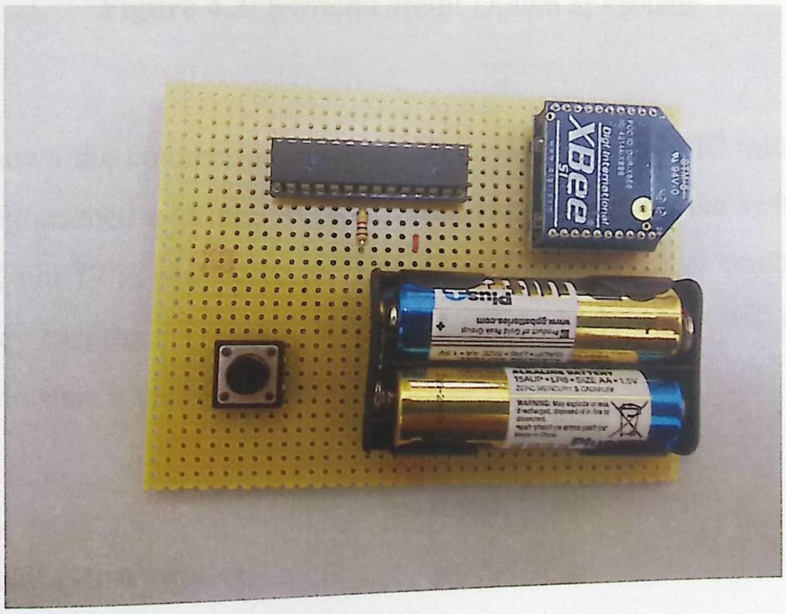


Figure 4.2: Hardware Switch Circuit

The circuit was drawn at Proteus as shown in figure 4.3:

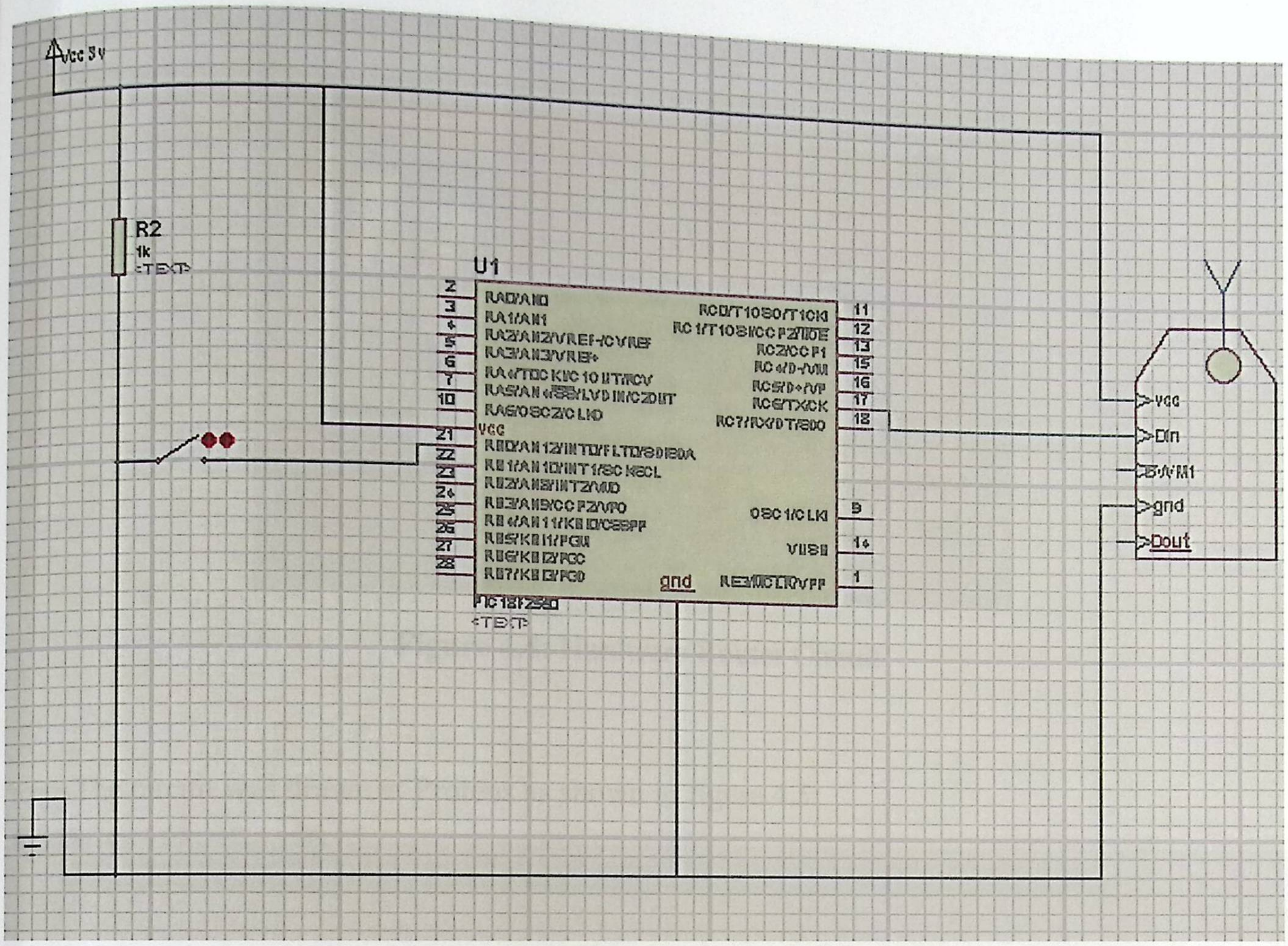


Figure 4.3: Switch Circuit Drawn at Proteus

This circuit shows the connection between XBee and 18f2550 PIC microcontroller, pin 21 of the PIC was programmed as a digital input to be connected with the switch that triggers the system, and the TX pin 17 is connected to DIN of the XBee to receive commands from PIC. So if there is emergency car, the driver will press the button which in turn will make the PIC sends command to XBee to start transmitting data.

- **Router circuit (street nodes):**

The circuit was built at board as shown in figure 4.4:

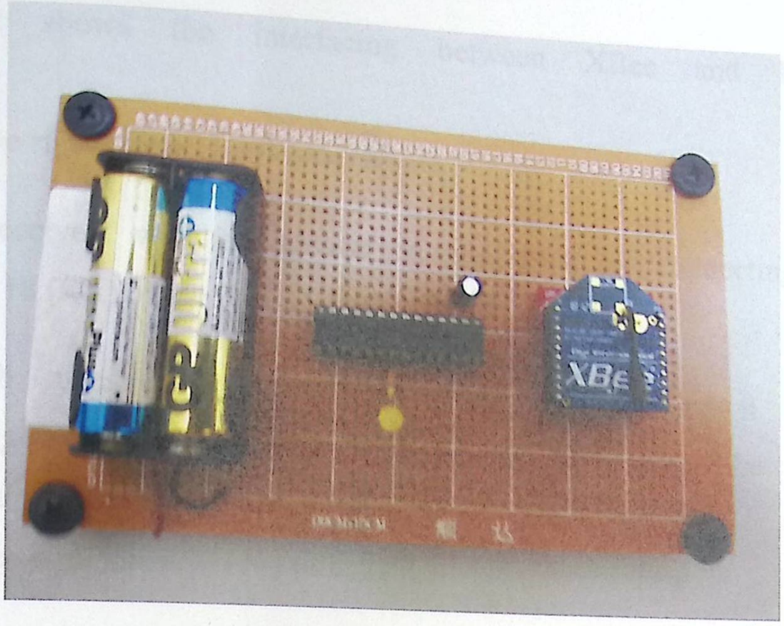


Figure 4.4: Hardware Router Circuit

The circuit was drawn at Proteus as shown in figure 4.5:

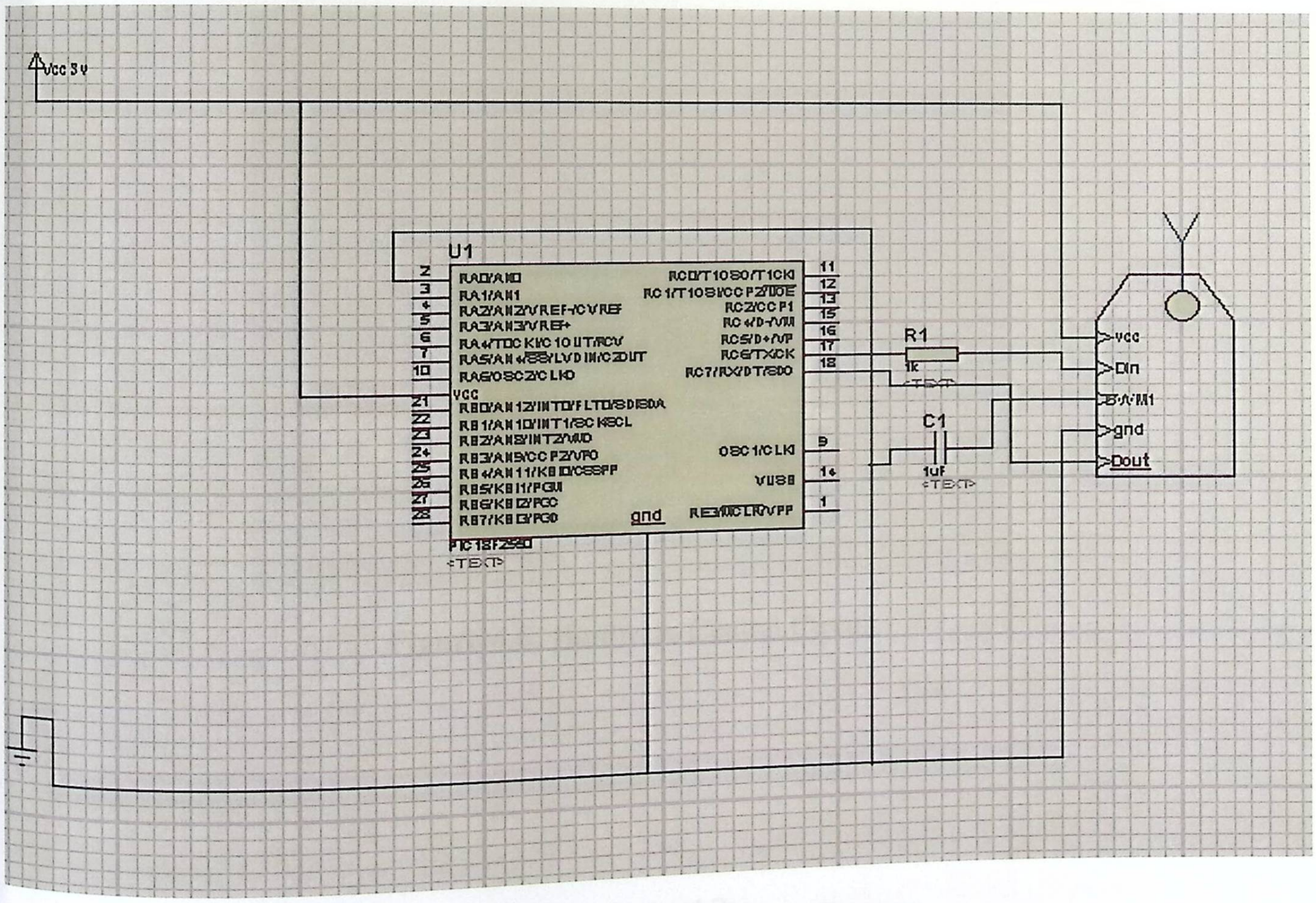


Figure 4.5: Router Circuit Drawn at Proteus

This circuit shows the interfacing between XBee and PIC 18f2550 as the following:

1) XBee will receive signal from coordinator XBee.

2) The signal received will be passed to the PIC by connecting pin DOUT of the XBee with pin 18 (RX) of the PIC.

3) The PIC will measure the power received and compare it with a threshold value, if it's larger than the threshold , the PIC will pass the XBee address to the XBee to be transmitted to the central XBee, and that's why we connect pin DIN of the XBee with pin 17 (TX) of the PIC.

Central circuit:

The circuit was built at board as below:

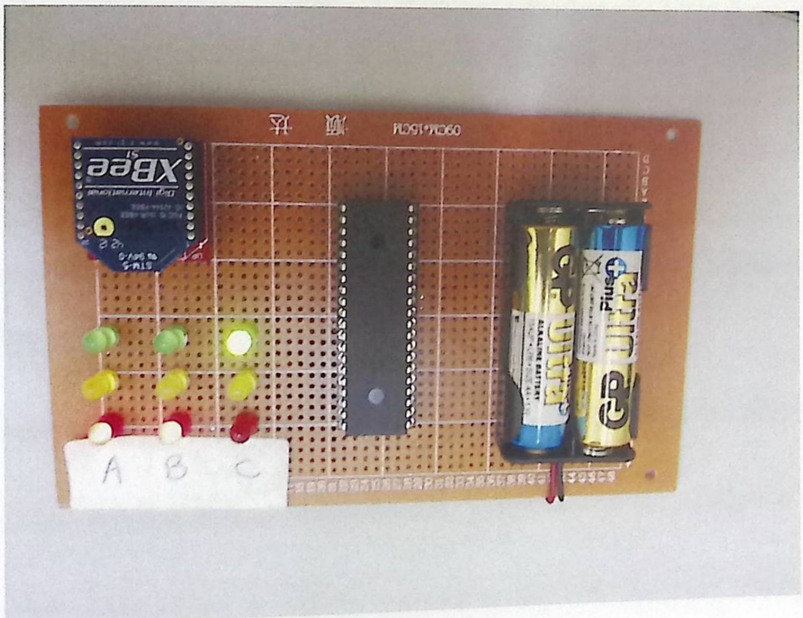


Figure 4.6: Hardware Central Circuit

The circuit was drawn at Proteus as shown in figure 4.7:

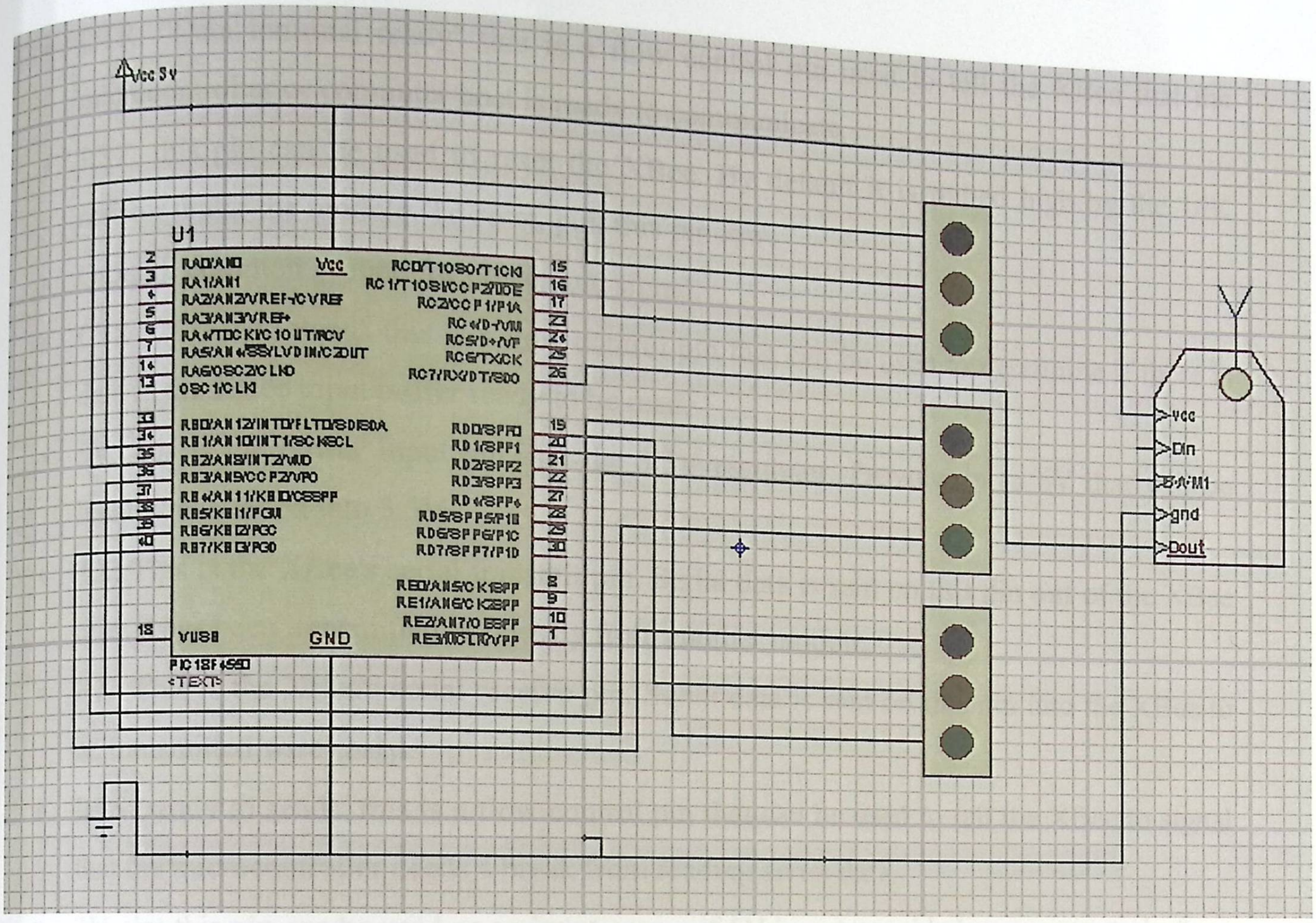


Figure 4.7: Central Circuit Drawn at Proteus

This circuit shows the interfacing between central XBee and PIC 18f4550 microcontroller as described in the following steps:

1) The traffic lights were connected to the PIC through port B from (b0-b7) and port D (d0).
2) Pin DOUT of the central XBee is connected to pin 26 (RX) of the PIC microcontroller.
3) When the signal is received, it will be passed to the PIC; to start the interrupt program so the PIC will change the normal behavior.

4.2.2 XBee interfacing

XBee ZB"Series1" will be used in this project. The XBee ZB RF Modules are designed to operate within the ZigBee protocol and support the unique needs of low-cost, low-power. The modules operate within the ISM 2.4 GHz frequency band.

- DTR: "Data terminal ready" this is a flow control pin used to tell the XBee that the microcontroller or computer host is ready to communicate.
- RST: this pin can be used to reset the XBee. By default it is pulled high by the 10K resistor under the module. To reset, pull this pin low.
- Ground: common ground for power and signal.
- CTS : "Clear to Send" this is a flow control pin that can be used to determine if there is data in the XBee input buffer ready to be read.
- 5V: this is the power input pin into the 3.3V regulator. Provide up to 6V that will be linearly converted into 3.3V
- TX: This is the XBee's serial transmit pin. Serial data is sent on this pin **out of** the XBee after it has been transmitted wirelessly from another module.
- RX: This is the XBee's serial receive pin. Serial data is sent on this pin into the XBee to be transmitted wirelessly.
- RTS (ready to send): this is a flow control pin that can be used to tell the XBee to signal that the microcontroller needs a break from reading serial data.
- 3V pin: this pin can be used as an input power if 5V is not provided or as output pin from 250mA regulator if 5V is provided [13].

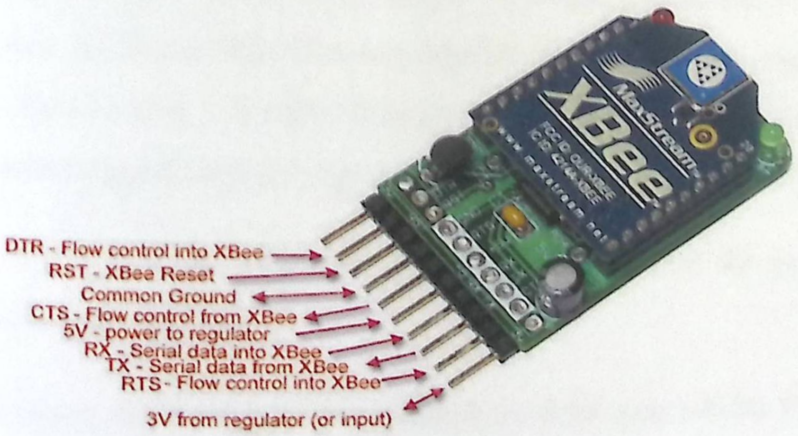


Figure 4.8: XBee Pins.

This figure shows the all connection of pins in every node which contains the PIC18F4550, XBee transceiver.

4.2.3 PIC microcontroller

Microcontroller is a Microchip programmable IC that controls the inputs and outputs from each device, it will be used as the main subject that controls the system for the goal to be achieved.

Two kinds of PICs will be used in this project, **PIC18F4550 and PIC18F2550**; the main difference between them is the number of pins. And here is an explanation for each one.

4.2.3.1 PIC18F4550 microcontroller:

PIC 18F4550 will be used because of its Nano Watt TECHNOLOGY PIC18F4450 incorporate a range of features that can significantly reduce power consumption during operation.

This PIC has 40 pins divided to 5 I/O ports (PORTA, PORTB, PORTC, PORTD and PORTE). PORTB and PORTD have 8 pins to receive/transmit 8-bit I/O data. The remaining ports have different numbers of pins for I/O data communications. The PIC has some pins just to get power, typically +5 volts and 0 volts or ground. Additionally there are 2 pins where a quartz crystal is attached to provide a basic "pulse" or clock for the PIC.PIC18F4550 has 12 different source for clock oscillator and there are internal clock oscillator consists from 8 user-selectable frequencies, from 31 kHz to 8 MHz Selection of particular internal frequency is obtained through specific programming of internal register which is called "OSCCON"[1].

This PIC was connected with the central XBee and the traffic light circuit so it can control them the way we want.

To program the device in this project in some pins of the PIC18F4550 can be used in different description as the following:

- **Pin 1 can be used as:**

  1) MCLR (Master Clear reset input): an input pin, it is an active-low pin which used to reset the device.
  2) Pp.: power pin which is a programming voltage input.
  3) RE3: this pin is used as Digital input.

- **Pin11**: (V_{DD}) power positive supply for logic and I/O pins.
- **Pin12**: (Vss) Ground reference for logic and I/O pins.
- Port B: from pin 33 to pin 40 are used as I/O digital data.
- Port D: that built from pin 19, 20, 21, 22, 27, 28, 29 and 30, are used as I/O digital data.
- **Pin 25 can be used as :**
  1) TX: EUSART asynchronous transmit.
  2) RC6 : for digital I/O
  3) CK: EUSART synchronous clock.

But in our circuit we used it as a transmitter.

- **Pin 26 can be used as :**
  1) RC7 : for digital I/O
  2) RX: EUSART asynchronous receive.
  3) DT: EUSART synchronous data (see TX/CK).
  4) SDO: SPI data out.

And in our circuit we used it as receiver.

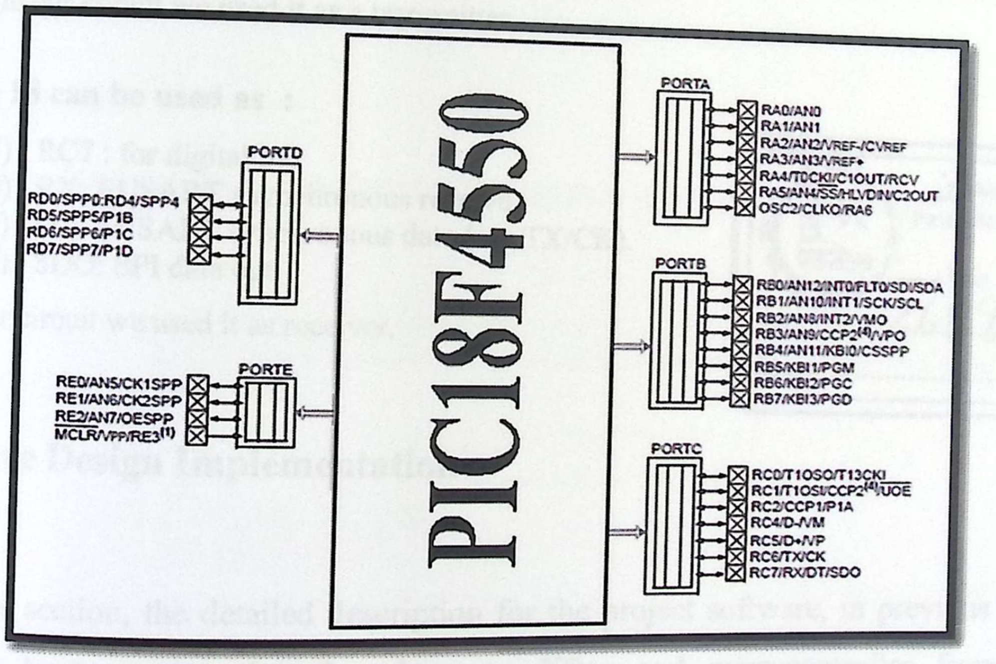


Figure 4.9: PIC18F4550 ADC Inputs

4.2.3.2 PIC18F2550 microcontroller:

This PIC was chosen also for the same properties of PIC18F4550, but less number of pins. This kind of PICs was connected to the XBee's on the street sides, where no ports need to be connected, and less number of pins needs to be used.

And here are the pins that were used in programming:

- **Pin 1 can be used as:**

    1) MCLR (Master Clear reset input): an input pin, it is an active-low pin which used to reset the device.
    2) Pp.: power pin which is a programming voltage input.
    3) RE3: this pin is used as Digital input.

- **Pin 8 or Pin 19**: (Vss) Ground reference for logic and I/O pins.
- **Pin 20**: (V_{DD}) power positive supply for logic and I/O pins.
- **Pin 17 can be used as:**
    1) TX: EUSART asynchronous transmit.
    2) RC6 : for digital I/O
    3) CK: EUSART synchronous clock.

    But in our circuit we used it as a transmitter

- **Pin 18 can be used as :**
    5) RC7 : for digital I/O
    6) RX: EUSART asynchronous receive.
    7) DT: EUSART synchronous data (see TX/CK).
    8) SDO: SPI data out.

And in our circuit we used it as receiver.

4.3 Software Design Implementation

In this section, the detailed description for the project software, in previous section we talked about how we can interface between XBee and microcontroller from hardware description, in this section we will describe software was used to have hardware interfaces goal. And for software implementation, Micro C was used to program microcontroller in C language.

4.3.1 PIC programming

Microcontroller acts as brain of the whole system. It receives the desired data from the XBee that interface with it .An algorithm is developed to make the microcontroller able to read the input and respond accordingly.

4.3.1.1 PIC code analysis

The PIC code is the code that is burned on the PIC in order to control the whole operation of the system, because the PIC is the core of this system.

In our system the PIC18f2550 perform three functions:

1) Communicate with the XBee that interface with it.
2) Getting the reading power from the XBee and converting these reading to volt.
3) Comparing these data with threshold value; and if the data is greater than the threshold, it sends command to the XBee to send the data to the central XBee.

And the PIC18f4550 perform two functions:

1) Controlling normal behavior of traffic light system.
2) Receiving the data from the central XBee and changing the behavior of the traffic light according these data.

4.3.1.2 MPLAB IDE

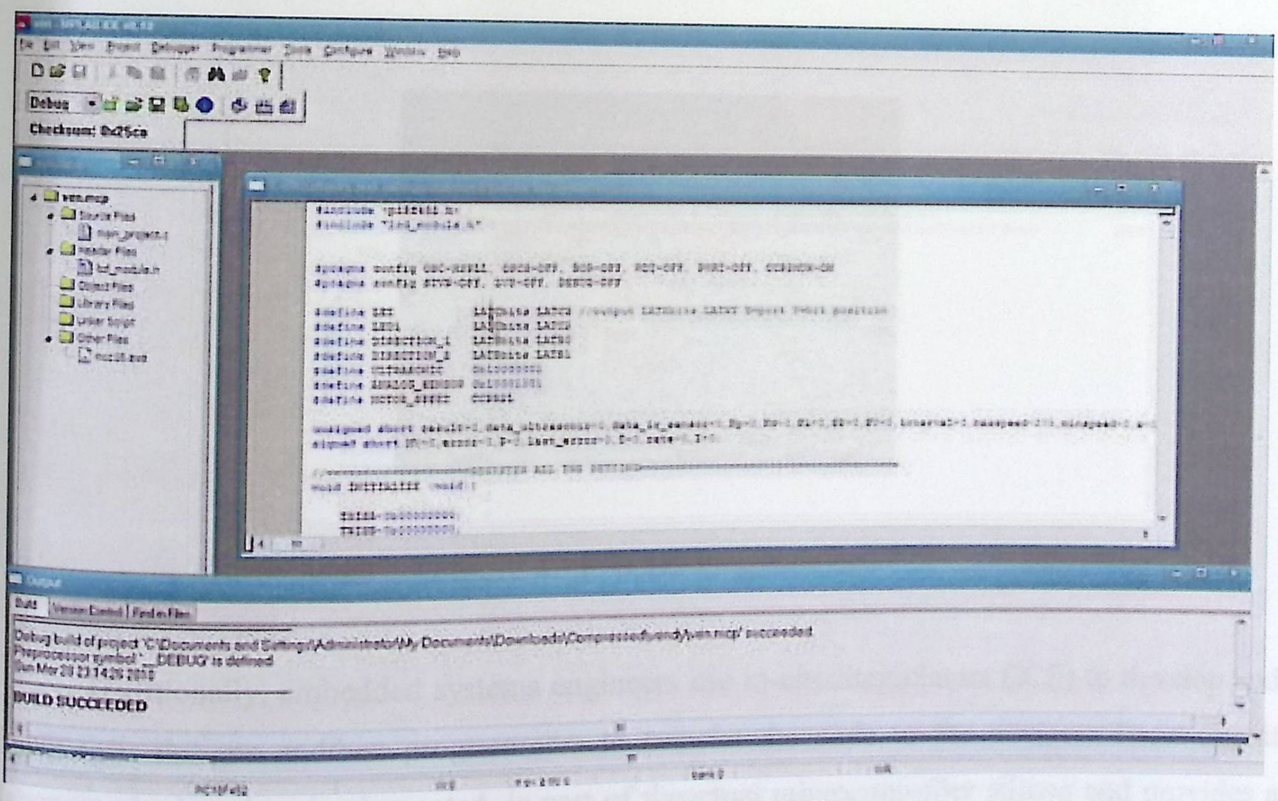


Figure 4.10: MPLAB IDE Editor

MPLAB IDE is a software program that runs on PC to provide a development environment for your embedded microcontroller design. Also it is a Windows based Integrated Development Environment for microchip Technology Incorporated PIC microcontroller (MCU) and dsPIC digital signal controller (DSC) families. MPLAB environment contains untitled workspace which contain needed file after create project, every needed file contain main four part Header file (p18F4550.h), Linker file (p18f4550_g.lkr), library file (p18f4550.lib), source file → add file that contain code of project . Output window for sure that code correctly executes.

In the MPLAB IDE, we can:

1) Create source code using the built-in editor.
2) Knowing which peripherals and pins control hardware, write the software. We use a compiler that allows a more natural language for creating programs.
3) Assemble, compile and link source code using various language tools. An assembler, linker and librarian come with MPLAB IDE.
4) "Burn" code into a microcontroller and verify that it executes correctly in finished application.

Figure 4.11: ICD2 Programmer

Traditionally, embedded systems engineers use in-circuitemulators (ICE) to develop and debug their designs andthen programmers to transfer the code to the devices. Thein-circuit debugging logic, when implemented, is part of theactual microcontroller silicon and provides a low-costalternative to a more expensive ICE.

Features

- USB (Full Speed 2 Mbits/s) and RS-232 interface to host PC
- Real-time execution
- MPLAB IDE compatible.
- Built-in over voltage/short circuit monitor
- Supports low voltage to 2.0 volts (2.0 to 6.0 ranges)
- Diagnostic LED's (Power, Busy, Error)
- Read/Write program and data memory of microcontroller
- Erase of program memory space with verification

4.3.1.3 Codes description:

Transmitting code:

The process of sending the emergency command is implemented through the following code:

```
Void main () {
OSCCON=0x72;
TRISA=1;
PORTA=0;
PORTB=0;
TRISB=1;

UART1_Init (9600);
while(1)
{

if(button(&PORTB,0,50,0))
{
UART1_Write_Text("Emergency");
delay_ms(5000);
}

} //close while loop
} //close void main
```

This code is used for sending an emergency message from the ZigBee transmitter to the routers that allocated on the streets when there is an emergency action.

Routing code:

The process of receiving an emergency command and taking the action according it is implemented through the following code:

```
char volt;
void main() {
OSCCON=0x72;
TRISA=1;
PORTA=0;
PORTB=0;
TRISB=0;
ADCON1=14;
UART1_Init(9600);
while(1)
{
volt=ADC_Read(0)/4;
if(volt>220)
{
delay_ms(3000);
UART1_Write_Text("*B#");
PORTB=1;
delay_ms(5000);
}
PORTB=0;
} //close while loop
} //close void main
```

The above code is used to check if the emergency vehicle near or far from the XBee that allocated on the street by:

✓ Measuring the power from the transmitter
✓ Converting it to voltage.
✓ Comparing it with threshold value .
✓ If it is greater than the threshold, XBee sends its address to the central XBee.

4.3.2 XBee configuration:

X-CTU software will be used to test and configure Bee module. The software is easy to use, update the parameters, and allows testing the radio modems in the actual environment with just a computer and the items included with the radio modems.

We can communicate with XBee through **"USB Serial Port".** This means that we have to stabilize a serial connection between the XBee board and PC device, With USB adapter.that is because we want to set the XBee parameters.

More than one device (XBee) can connect on serial port of the PC; so that we can test the wireless communication and connect more than one XBee to our PC.

We interested in these four important basic configurations on the XBee:

- baud rate for serial communication
- network identifier
- node address
- destination node address

There are three different XBee configarations , one for the coerdinatore,router , end device (central).

- The coerdinatore XBee configaration :

Step 1:

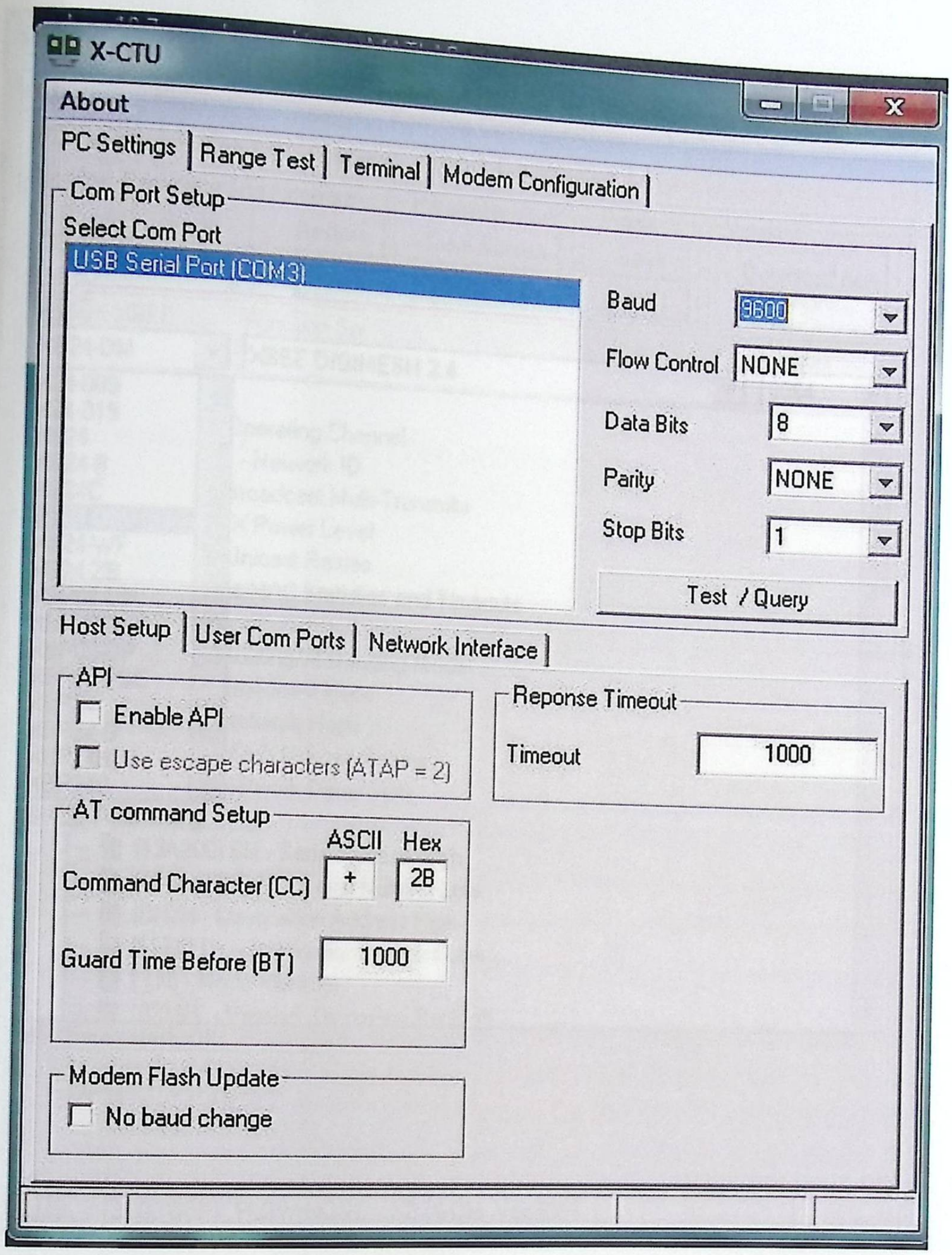


Figure 4.12: PC setting

Select com port in section : "PC Settings"

1) Baud Rate set to 9600 .
2) Flow Control : NONE
3) Data Bits : 8
4) Parity : NONE
5) Stop Bits : 1
6) Click "Test / Query" , so the XBee serial address will be shown.

Step 2:

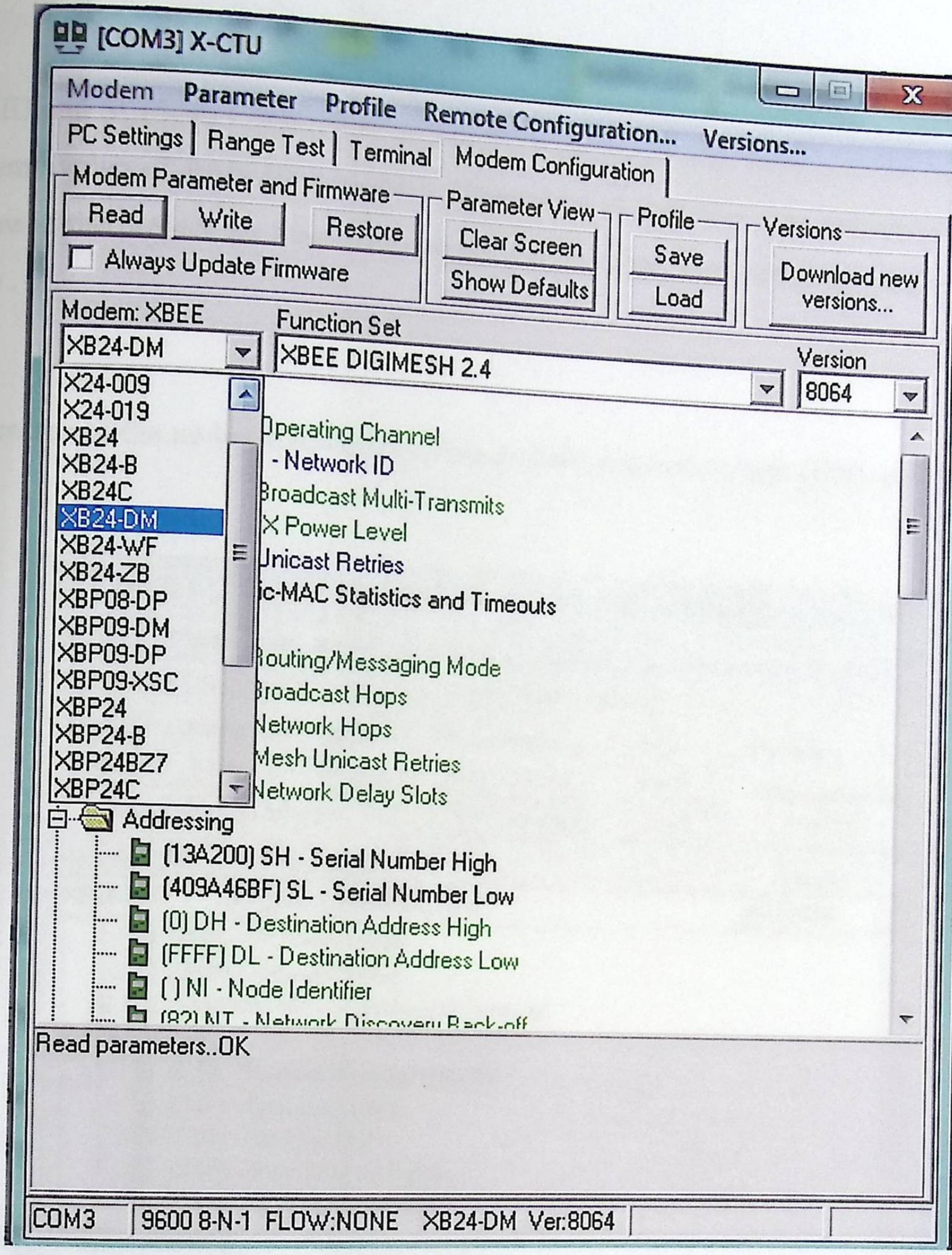


Figure 4.13: Setting The Modem Configuration

Go To Modem Configuration and click the Read Button , then start configure it as the following :

1) select the" **modem**" type" XB24-"DM".

2) **"Function set"** is the firmware inside XBee. It is responsible for different topogies and configuration. In this project the function is set as XBEE DIGIMESH 2.4

3) The **"Version "** is set 8064.

4) Click **"Write"** to load the new firmware.

Step 3:

Set PAN ID and SC(Scan Channel) to an arbitrary value. All the devices in the network should use the same value of PAN ID and same channel of SC, so they can communicate with each others.Here in the project we keep them as the defult value, that means we don't change it to a new value .

Step 4:

Set the adresses of the node (distination), that include distination high (DH) and distination low (DL)

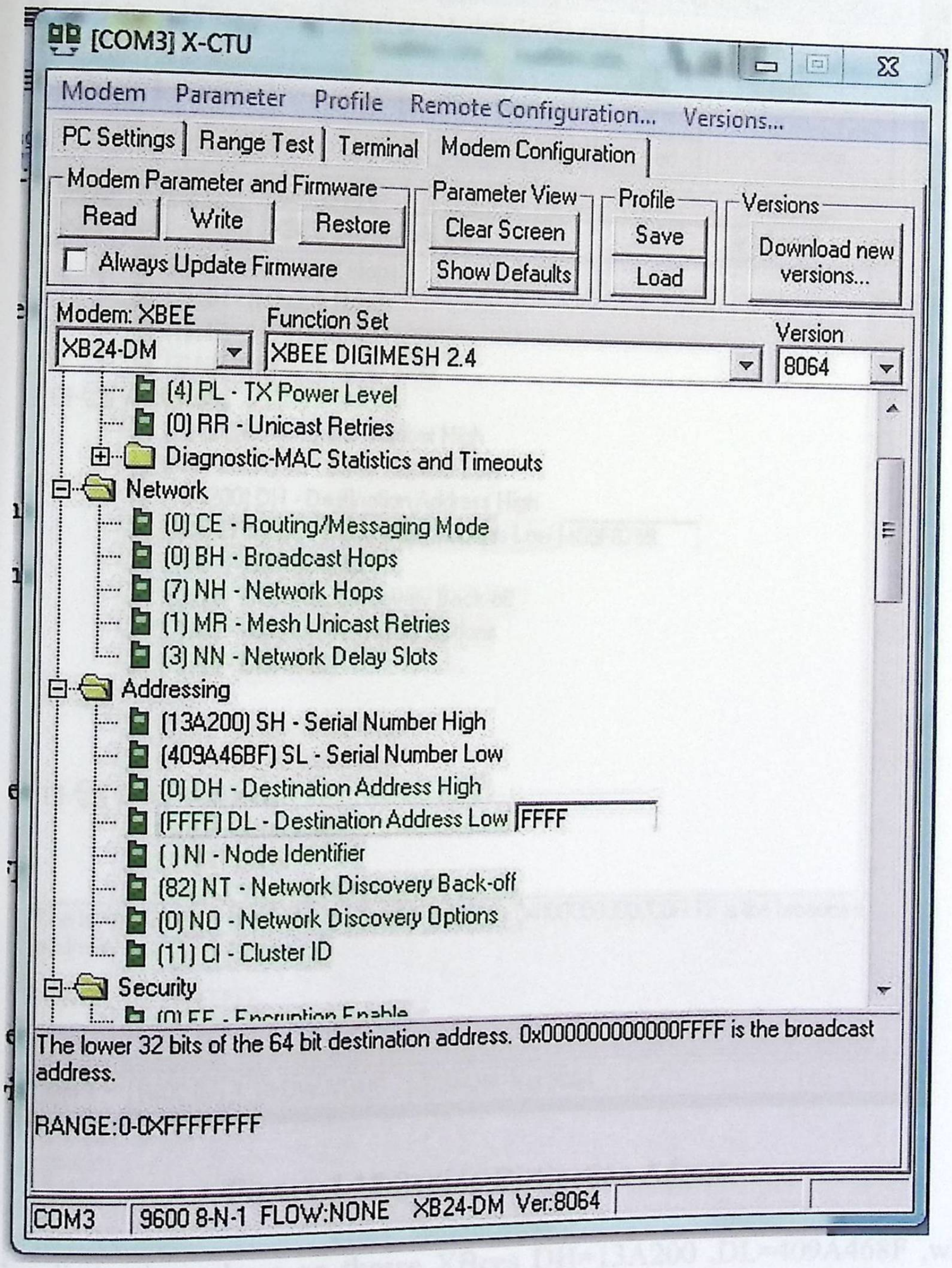


Figure 4.14: Setting The Modem Configuration For The Coordinato

The coordinator XBee that set on the emergency vehicle will send data as broadcast, so the DH will be set 0000 and DL will be set FFFF.

- The router XBee configaration :

The router XBee that is set on the four street sids of the intersection, to configure these XBees we did the same steps as shown before , with one different is the DH and DL, as shown in the figure.

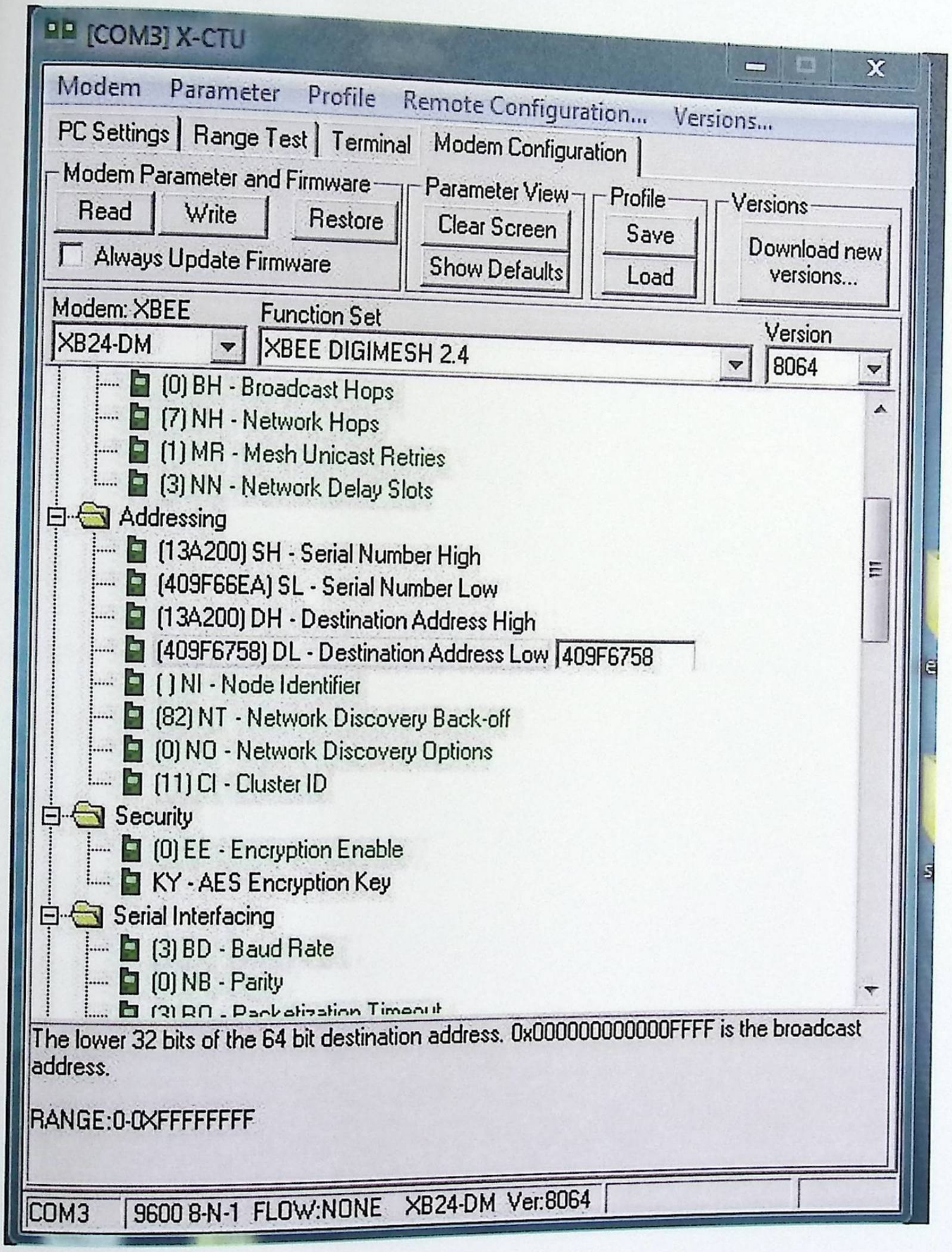


Figure 4.15:Setting Distination Adress

Set the distination adress on thesse XBees DH=13A200 .DL=409A468F ,which is the serial adress of the end device (the central XBee).

- The end device configaration :

Same as before but without configurate the DH and DL. Because it don't need to transmite any data to any devices , it just receive data and pass it to the PIC microcontroller.

5

Chapter Five

Testing and Results

5.1 Introduction

5.2 Testing and Results

5.2.1 XBEE testing

5.2.2 PIC Testing

5.3 Performance evaluation

5.3.1 Power measurements

5.3.2 Delay

5.3.3 Localization

5.3.4 Performance analysis

5.1 Introduction

The final stage to complete the project is to test the system to get results and measure the performance of our system. This chapter shows all measurements needed to evaluate the performance of this system such as delay, power, and localization.

5.2 Testing

Checking and testing of the microcontrollers and XBee transceivers are done in this section including the testing results.

5.2.1 XBee testing

Here we test the connection between the coordinator XBee and the XBees that set on the street sides (routers).

The coordinator XBee will send data that include the word "EMERGENCY" to the routers, and as shown this is what we get at the router XBees.

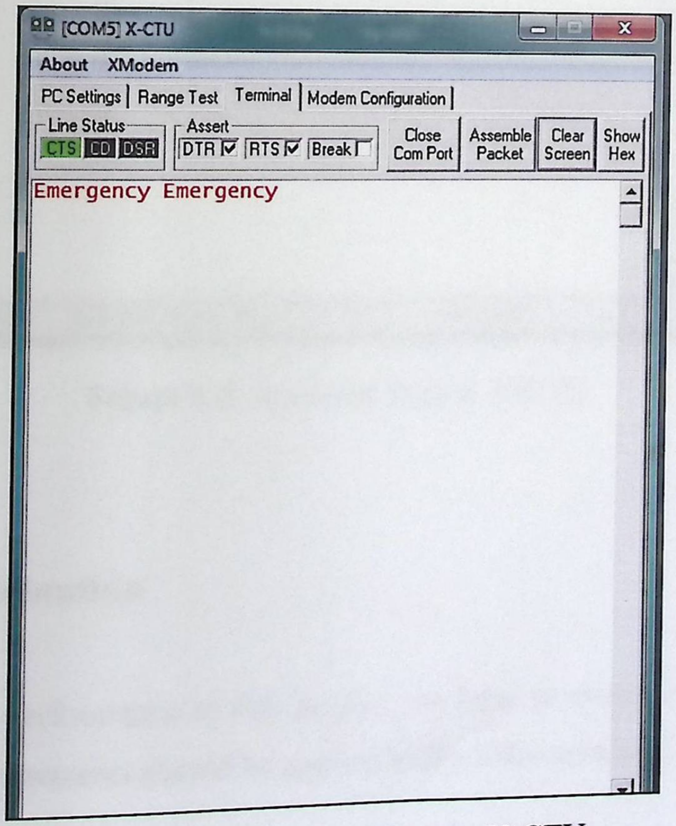


Figure 5.1: Received Data at X-CTU

5.2.2 PIC microcontroller testing

The PICs that were connected to the routers will measure the power of the received signal from the coordinator, and compare it with a threshold value, if it exceeds it, the PIC will pass the XBEE address to the router to transmit it to the central XBee.

To make sure that the PICs can communicate and exchange data with the XBees, we test what we received at the central XBee which is the address of the router as shown.

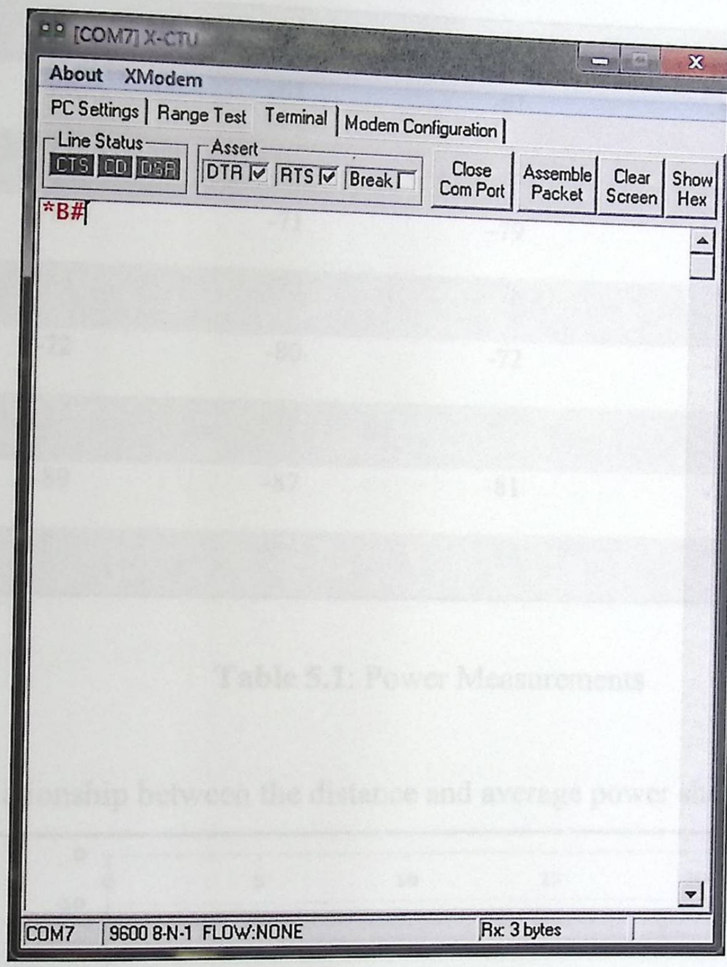


Figure 5.2: Received Data at X-CTU

5.3 Performance evaluation

To evaluate the performance in this project, we have to measure the delay, power and localization. These measurements should be applied to the XBee scenario.

5.3.1 Power measurements:

The power measurement is a very important factor in our project, for determining the threshold value that will be used in the router side, so for choosing the good threshold value we take some values of power with respect to change in distance, as shown.

Distance (m)	Power1 (dBm)	Power2 (dBm)	Power3 (dBm)	Power4 (dBm)	Avg Power (dBm)
0	-48	-51	-49	-47	-48.75
2	-58	-61	-61	-55	-58.75
4	-71	-65	-72	-71	-69.75
6	-74	-71	-79	-75	-74.75
10	-83	-78	-78	-78	-75.5
12	-72	-80	-72	-81	-76.25
14	-85	-78	-78	-82	-80.75
18	-89	-87	-81	-76	-85.25
20	-88	-88	-82	-86	-88.75

Table 5.1: Power Measurements

The relationship between the distance and average power shown in the curve:

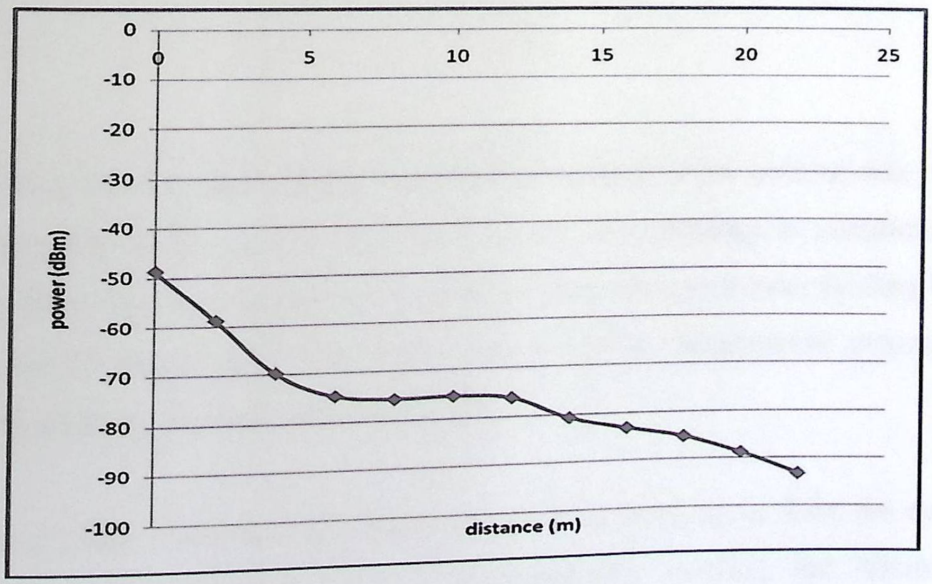


Figure 5.3: Relation between Distance and Average Power

The measurements were taken by X_CTU program with commands as shown in the figure below:

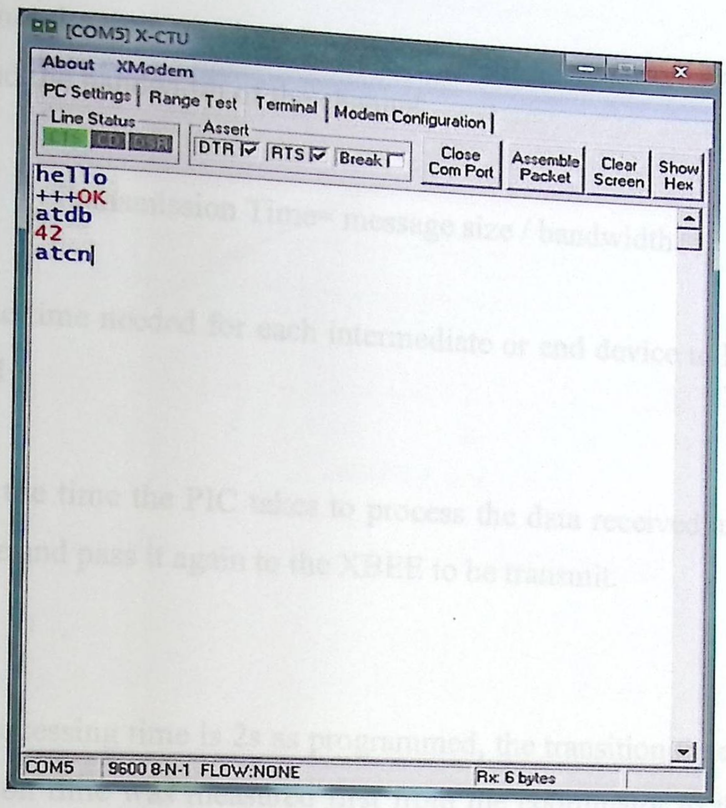


Figure 5.4: Power Measurments In X-CTU

First we send any data such as “hello” then we access the AT command and measure the power using the command ATDB.

And this measure was taken on distance 6m, the power reading is 42 dBm in hex, which = 66 dBm in decimal.

5.3.2 Delay

The delay is an important factor in this project as the nature of the sending data requires fast transmission. The delay defines how long it takes for an entire message to completely arrive at the central XBee from the time the first bit is sent out from the coordinator passing through the routers that sets on the streets sides. The delay is made of four components: propagation time, transmission time, queuing time and processing delay.

1) Propagation Time: measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

Propagation Time=Distance/ propagation speed

2) Transmission Time: the time required for transmission of a massage depends on the size of the message and the bandwidth of the channel.

Transmission Time= message size / bandwidth

3) Queuing Time: the time needed for each intermediate or end device to hold the message before it can be processed

4) Processing Time: the time the PIC takes to process the data received and compare it to the threshold value and pass it again to the XBEE to be transmit.

In our project the processing time is 2s as programmed, the transition time as measured is 2.5ms .and the propagation time was measured first from the coordinator to the router, it was taken many times with changing in distance as shown in the table , second from the router to the central and this is affixed value.

Distance in (m)	Delay in (ns)
2	6
4	13
6	20
8	26
10	33
12	40

Table 5.2: Delay From Coordinator To router

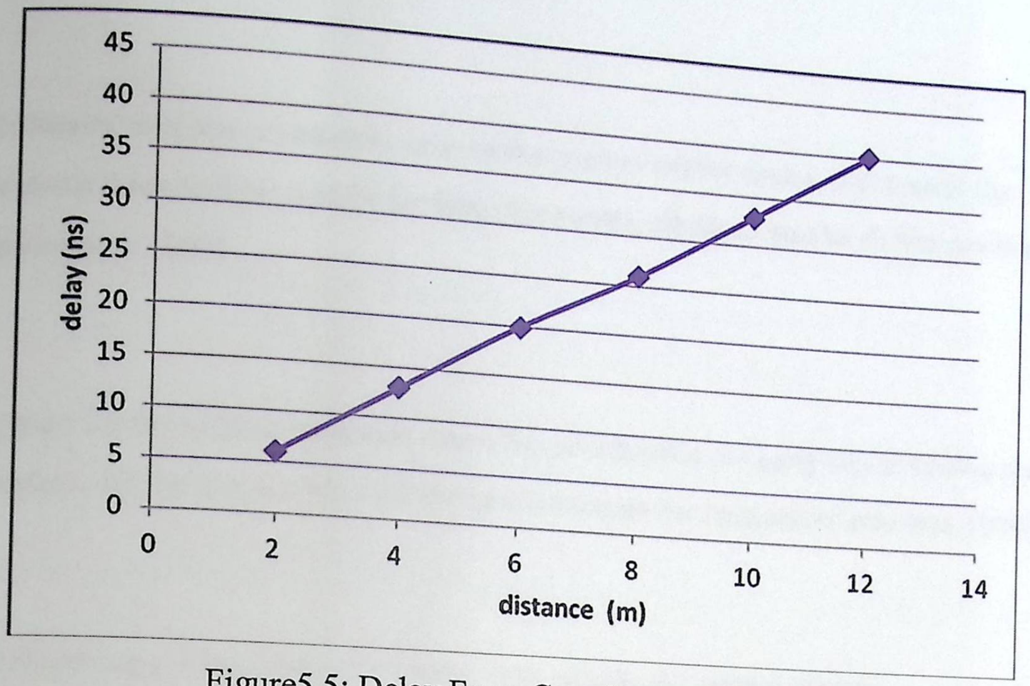


Figure5.5: Delay From Coordinator To Router

5.3.3 Localization:

Localization should be taken carefully, XBees on the streets must be localized in a way that avoids the interference between XBees.

XBees has been placed on a certain distance from the traffic signal, the signal was sent from the ambulance, and the nearest XBee received the signal & send it to the central XBee.

The better distance between XBee's on the streets & the traffic light to get right results was 23m at this distance we avoided interference between them & we got our goal which is to know the street that the ambulance will come from.

5.3.4 Performance analysis:

We applied the project in B+ park, in many cases, at the different environments, and this is the results we got:

Case 1:

When the coordinator was approximately near to the routers (street nods), that means the coordinator is about three to four meters far from the router, all times and in all the environments the successive rate was 100% .

Case 2:

Repeat all the tests for the system response when the coordinator is nearly ten to twelve meters far from the routers, all the times and in all the environments the successive rate was 100%.

Case 3:

This was the critical case, where the coordinator was near to the central circuit.

For explanation : if the emergency vehicle is coming from street "A", and the driver trigger the system when he was standing on the traffic light, the successive rat shows the percentage of how many times the light of street "A" is set as green, and close the others, and the error rate shows the percentage of how many times the light of street "B" or "C" is set as green instead of street "A".

The total results we got are shown below:

- In the morning at 8:00am, when there is no movement; no students, no cars. We found that after15 trials, 13 were succeeded and 2 were failed.

  By calculations, successive rate= 13/15= 86.66%.

  Error rate = 13.33%.

- In the middle of the day, at 12:30 pm, where there was movement of students and cars, we found that after 15 trials, 12 were succeeded and 3were failed.

  By calculations, successive rate= 14/18= 80%.

  Error rate = 20%.

- In a windy day, and after 15 trials, 10 were succeeded and 5were failed.

  By calculations, successive rate= 10/15= 66.6%.

  Error rate = 33.3%.

These results in case three were unacceptable, so we tried to reduce the error percentage by reducing the threshold value, and re-localizing the street nodes. So we improve the results as the following:

- In the middle of the day, at 12:15pm, where there was movement of students and cars, we found that after 15 trials, 13 were succeeded and 2 were failed.
  By calculations, successive rate= 13/15= 86.6%.

Error rate = 13.3%.

We aspired to improve the result we got at the windy day, but unfortunately we couldn't take it because there were no more windy days .
But as expecting the results will be better than before, as we got at the previous experiment .

6

Chapter Six

Recommendations and Conclusion

6.1 Introduction

6.2 System achievements

6.3 Real learning outcomes

6.4 Recommendations

6.5 Conclusions

6.1 Introduction

The project has been done step by step, for developing a new idea which is building a smart traffic light.

The project was a good step in implementing a system that controls the traffic light by the emergency vehicle. Meanwhile we have some recommendations and suggestions for the future work that can be taken to make the system more efficient and economic.

6.2 System achievements

Almost all the goals of our system have been achieved. In this point the main achievements of the system are discussed and the ways of achieving it.

We build a traffic light system that approximately similar to the system in our city by programming the microcontroller.

We configure the XBee transceiver, we configure the transmitter one to broadcast signal to reach XBees that allocated on the roads & we configure the XBee's on roads to routing data to central XBee to pass signal to the microcontroller to control traffic lights.

We build a coordinator node near the traffic light that consists of XBee transceiver and make a serial communication between PIC and this XBee to pass data between them.

We control the traffic light system according to this data; we can run our application continuously

6.3 Real learning outcomes

After the implementation of the project we have an expert in the following points:

- Learn how to build traffic light system
- Learn how to use and program 18F4550 microcontroller.
- XBee configuration.
- Learn how to create network using ZigBee technology and make routing.

- Learn how to solve many problems that we face in ZigBee communications.

6.4 Recommendations

After we worked on this project, and faced many problems during the implementation, we saw that the following points may be a good improvement for this project in order to make it more reliable:

- An improvement to the system could be applied, by using sensors on the traffic light to be more accurate in deciding that the ambulance was passed the traffic light.
- Measuring the load in each street in the intersection, and give the street with the highest load the priority of the green light.

6.5 Conclusion

Finally at the end of the long work in this project, we built the smart traffic light system, and it has efficiently worked in the way we want. We also solve the problems and challenges that we faced.

The general challenges include how to deal with three software programs which is Proteus, micro c and X-CTU, another challenge was how to collect all the hardware's and built the circuits and the biggest challenge was in the first moment we turn on the system, and start finding the way to solve the critical case, which is how to decide the street that the emergency vehicle comes from in case the driver trigger the system when he was standing on the traffic light.

And the specific challenges were:

1) The power reading was changed according to the environment changes.
2) Determining the exact gray region (critical region) which is the region where the error appears.
3) The ability of determining the street where the emergency vehicle comes from.

But at the end of the day it was a nice thing to see our dream and effort in building this system becomes true, also it was an interesting work in this group.

And finally in few words …..

Souls of our lovers are the most important thing of our life, nothing can compensate their absence, & regret will not change anything and it will be too late if we didn't do something for them, so this project was built to save their souls by allowing the emergency vehicle to reach the place it need in due course and without problems in heavy traffic.

References

[1] http://networkedblogs.com/hH0DU

[2] http://1000projects.org/traffic-controlling-system-for-ambulance.html

[3] http://www.authorstream.com/Presentation/karthik122-1311778-intelligent-traffic-light control-for-ambulance/

[4] Zhang Yuye & Yan Weisheng, (2009) "Research of Traffic Signal Light Intelligent Control System Based On Microcontroller", First International Workshop on Education Technology and Computer Science,pp301- 303.

[5] Ahmed S. Salama, Bahaa K. Saleh & Mohamad M. Eassa (2010) "Intelligent Cross Road Traffic Management System (ICRTMS)", 2nd International Conference on Computer Technology and Development, pp27- 31.

[6] and [7] Comparison of the IEEE 802.11, 802.15.1, 802.15.4 and 802.15.6 wireless standards by "Jan Mange Tjensvold".

[8] Getting Started with ZigBee and IEEE 802.15.4, http://www.daintree.net

[9] Extending 2.4 GHz ZigBee® Short-Range Radio Performance with Skyworks SKY65344/SKY65343 Front-End Modules

[10] The meaning of traffic lights/red green and amber robots – South Africa. Mydriverslicence.co.za.

[11] Fascinating facts about the invention of the **Traffic Light** by **William L. Potts** in **1920** http://www.ideafinder.com/history/inventions/trafficlight.htm

[12] Track signal design http://www.cdeep.iitb.ac.in/nptel/Civil%20Engineering/Transportation%20Engg%201/41-Ltexhtml/nptel_ceTEI_L41.pdf

[13] XBee®/XBee-PRO® RF Modules

'Product Manual v1.xEx - 802.15.4 Protocol For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001'

Appendix A

PIC18F4550

Microcontroller

Datasheet

PIC18F4550 Datasheet Snapshots

MICROCHIP

PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 µA typical
- Sleep mode currents down to 0.1 µA typical
- Timer1 Oscillator: 1.1 µA typical, 32 kHz, 2V
- Watchdog Timer: 2.1 µA typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns ($T_{CY}/16$)
  - Compare is 16-bit, max. resolution 83.3 ns (T_{CY})
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I²C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

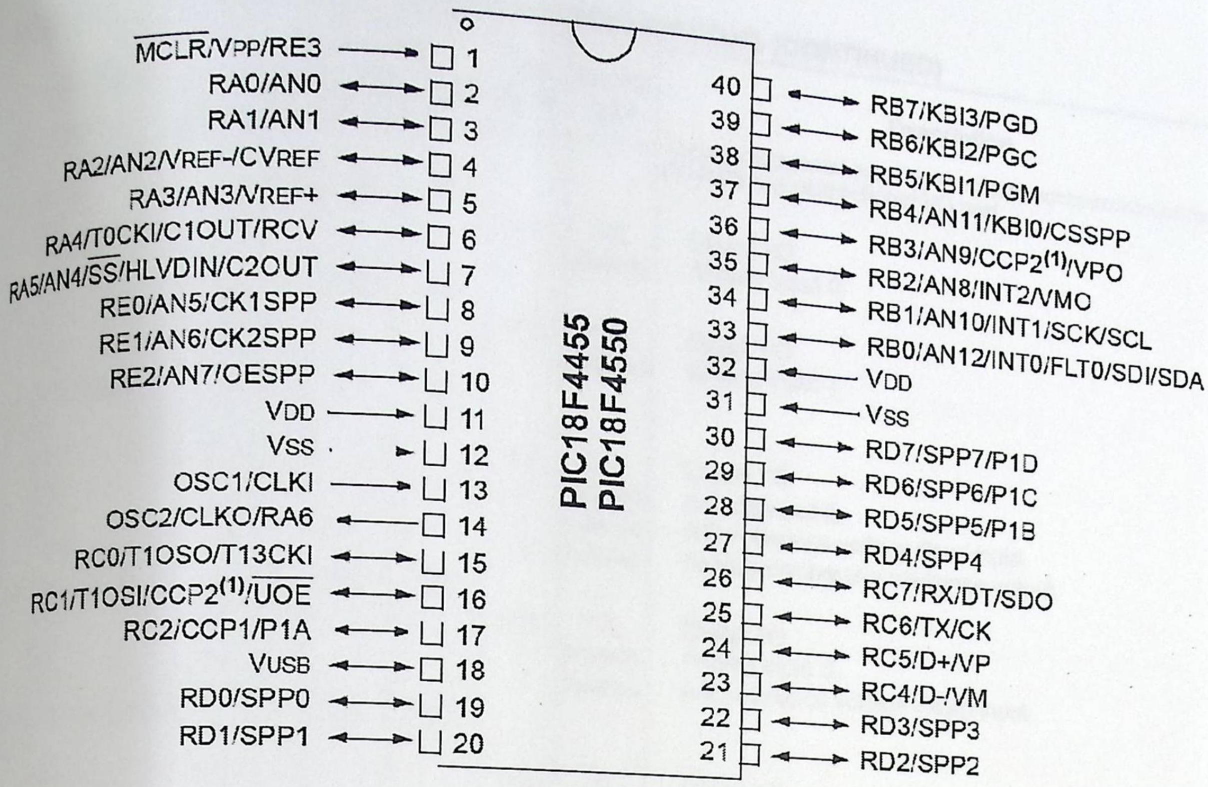


TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
MCLR/VPP/RE3	1	18	18			Master Clear (input) or programming voltage (input).
MCLR				I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
VPP				P		Programming voltage input.
RE3				I	ST	Digital input.
OSC1/CLKI	13	32	30			Oscillator crystal or external clock input.
OSC1				I	Analog	Oscillator crystal input or external clock source input.
CLKI				I	Analog	External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pin.)
OSC2/CLKO/RA6	14	33	31			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	—	In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6				I/O	TTL	General purpose I/O pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels; I = Input
O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.
2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number PDIP	QFN	TQFP	Pin Type	Buffer Type	Description
						PORTA is a bidirectional I/O port.
RA0/AN0	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-/CVREF	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
CVREF				O	Analog	Analog comparator reference output.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI/C1OUT/RCV	6	23	23			
RA4				I/O	ST	Digital I/O.
T0CKI				I	ST	Timer0 external clock input.
C1OUT				O	—	Comparator 1 output.
RCV				I	TTL	External USB transceiver RCV input.
RA5/AN4/$\overline{SS}$/HLVDIN/C2OUT	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
$\overline{SS}$				I	TTL	SPI slave select input.
HLVDIN				I	Analog	High/Low-Voltage Detect input.
C2OUT				O	—	Comparator 2 output.
RA6	—	—	—	—	—	See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels; I = Input
O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.
2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the $\overline{DEBUG}$ Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN12/INT0/FLT0/SDI/SDA	33	9	8			
RB0				I/O	TTL	Digital I/O.
AN12				I	Analog	Analog input 12.
INT0				I	ST	External interrupt 0.
FLT0				I	ST	Enhanced PWM Fault input (ECCP1 module).
SDI				I	ST	SPI data in.
SDA				I/O	ST	I^2C™ data I/O.
RB1/AN10/INT1/SCK/SCL	34	10	9			
RB1				I/O	TTL	Digital I/O.
AN10				I	Analog	Analog input 10.
INT1				I	ST	External interrupt 1.
SCK				I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL				I/O	ST	Synchronous serial clock input/output for I^2C mode.
RB2/AN8/INT2/VMO	35	11	10			
RB2				I/O	TTL	Digital I/O.
AN8				I	Analog	Analog input 8.
INT2				I	ST	External interrupt 2.
VMO				O	—	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	36	12	11			
RB3				I/O	TTL	Digital I/O.
AN9				I	Analog	Analog input 9.
CCP2(1)				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
VPO				O	—	External USB transceiver VPO output.
RB4/AN11/KBI0/CSSPP	37	14	14			
RB4				I/O	TTL	Digital I/O.
AN11				I	Analog	Analog input 11.
KBI0				I	TTL	Interrupt-on-change pin.
CSSPP				O	—	SPP chip select control output.
RB5/KBI1/PGM	38	15	15			
RB5				I/O	TTL	Digital I/O.
KBI1				I	TTL	Interrupt-on-change pin.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	Digital I/O.
KBI2				I	TTL	Interrupt-on-change pin.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	Digital I/O.
KBI3				I	TTL	Interrupt-on-change pin.
PGD				I/O	ST	In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the $\overline{DEBUG}$ Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	15	34	32			
RC0				I/O	ST	Digital I/O.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/$\overline{UOE}$	16	35	35			
RC1				I/O	ST	Digital I/O.
T1OSI				I	CMOS	Timer1 oscillator input.
CCP2(2)				I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
$\overline{UOE}$				O	—	External USB transceiver $\overline{OE}$ output.
RC2/CCP1/P1A	17	36	36			
RC2				I/O	ST	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
P1A				O	TTL	Enhanced CCP1 PWM output, channel A.
RC4/D-/VM	23	42	42			
RC4				I	TTL	Digital input.
D-				I/O	—	USB differential minus line (input/output).
VM				I	TTL	External USB transceiver VM input.
RC5/D+/VP	24	43	43			
RC5				I	TTL	Digital input.
D+				I/O	—	USB differential plus line (input/output).
VP				I	TTL	External USB transceiver VP input.
RC6/TX/CK	25	44	44			
RC6				I/O	ST	Digital I/O.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO	26	1	1			
RC7				I/O	ST	Digital I/O.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see TX/CK).
SDO				O	—	SPI data out.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the $\overline{DEBUG}$ Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTD is a bidirectional I/O port or a Streaming Parallel Port (SPP). These pins have TTL input buffers when the SPP module is enabled.
RD0/SPP0	19	38	38			
RD0				I/O	ST	Digital I/O.
SPP0				I/O	TTL	Streaming Parallel Port data.
RD1/SPP1	20	39	39			
RD1				I/O	ST	Digital I/O.
SPP1				I/O	TTL	Streaming Parallel Port data.
RD2/SPP2	21	40	40			
RD2				I/O	ST	Digital I/O.
SPP2				I/O	TTL	Streaming Parallel Port data.
RD3/SPP3	22	41	41			
RD3				I/O	ST	Digital I/O.
SPP3				I/O	TTL	Streaming Parallel Port data.
RD4/SPP4	27	2	2			
RD4				I/O	ST	Digital I/O.
SPP4				I/O	TTL	Streaming Parallel Port data.
RD5/SPP5/P1B	28	3	3			
RD5				I/O	ST	Digital I/O.
SPP5				I/O	TTL	Streaming Parallel Port data.
P1B				O	—	Enhanced CCP1 PWM output, channel B.
RD6/SPP6/P1C	29	4	4			
RD6				I/O	ST	Digital I/O.
SPP6				I/O	TTL	Streaming Parallel Port data.
P1C				O	—	Enhanced CCP1 PWM output, channel C.
RD7/SPP7/P1D	30	5	5			
RD7				I/O	ST	Digital I/O.
SPP7				I/O	TTL	Streaming Parallel Port data.
P1D				O	—	Enhanced CCP1 PWM output, channel D.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels; I = Input
O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.
2: Default assignment for CCP2 when CCP2MX Configuration bit is set.
3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
						PORTE is a bidirectional I/O port.
RE0/AN5/CK1SPP	8	25	25			
RE0				I/O	ST	Digital I/O.
AN5				I	Analog	Analog input 5.
CK1SPP				O	—	SPP clock 1 output.
RE1/AN6/CK2SPP	9	26	26			
RE1				I/O	ST	Digital I/O.
AN6				I	Analog	Analog input 6.
CK2SPP				O	—	SPP clock 2 output.
RE2/AN7/OESPP	10	27	27			
RE2				I/O	ST	Digital I/O.
AN7				I	Analog	Analog input 7.
OESPP				O	—	SPP output enable output.
RE3	—	—	—	—	—	See $\overline{MCLR}$/VPP/RE3 pin.
VSS	12, 31	6, 30, 31	6, 29	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	7, 8, 28, 29	7, 28	P	—	Positive supply for logic and I/O pins.
VUSB	18	37	37	O	—	Internal USB 3.3V voltage regulator output.
NC/ICCK/ICPGC(3)	—	—	12			No Connect or dedicated ICD/ICSP™ port clock.
ICCK				I/O	ST	In-Circuit Debugger clock.
ICPGC				I/O	ST	ICSP programming clock.
NC/ICDT/ICPGD(3)	—	—	13			No Connect or dedicated ICD/ICSP port clock.
ICDT				I/O	ST	In-Circuit Debugger data.
ICPGD				I/O	ST	ICSP programming data.
NC/$\overline{ICRST}$/ICVPP(3)	—	—	33			No Connect or dedicated ICD/ICSP port Reset.
$\overline{ICRST}$				I	—	Master Clear (Reset) input.
ICVPP				P	—	Programming voltage input.
NC/ICPORTS(3)	—	—	34	P	—	No Connect or 28-pin device emulation.
ICPORTS						Enable 28-pin device emulation when connected to Vss.
NC	—	13	—	—	—	No Connect.

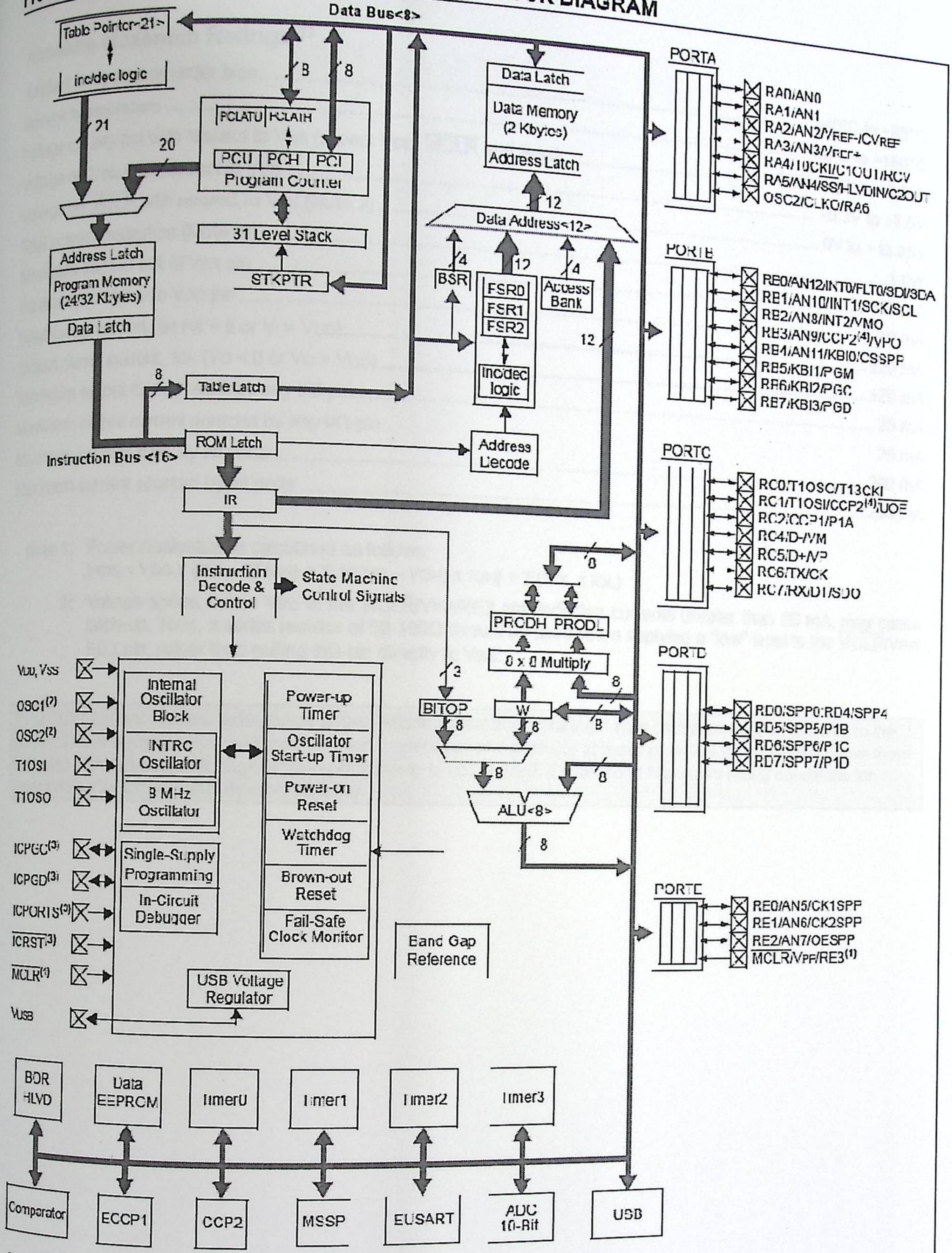
Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the $\overline{ICPRT}$ Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the $\overline{DEBUG}$ Configuration bit is cleared.

FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM



Note 1: RE3 is multiplexed with $\overline{MCLR}$ and is only available when the $\overline{MCLR}$ Resets are disabled.

2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Configurations"** for additional information.

3: These pins are only available on 44-pin TQFP packages under certain conditions. Refer to **Section 25.9 "Special ICPORT Features (Designated Packages Only)"** for additional information.

4: RB3 is the alternate pin for CCP2 multiplexing.

28.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(†)

Ambient temperature under bias .... -40°C to +85°C
Storage temperature .... 65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD}, $\overline{MCLR}$ and RA4) .... -0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS} .... -0.3V to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} **(Note 2)** .... 0V to +13.25V
Total power dissipation **(Note 1)** 1.0W
Maximum current out of V_{SS} pin .... 300 mA
Maximum current into V_{DD} pin .... 250 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) .... ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD}) .... ±20 mA
Maximum output current sunk by any I/O pin .... 25 mA
Maximum output current sourced by any I/O pin .... 25 mA
Maximum current sunk by all ports .... 200 mA
Maximum current sourced by all ports .... 200 mA

Note 1: Power dissipation is calculated as follows:
$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

2: Voltage spikes below V_{SS} at the $\overline{MCLR}$/V_{PP}/RE3 pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$/V_{PP}/RE3 pin, rather than pulling this pin directly to V_{SS}.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Appendix B

Codes

Central PIC code :

```
const unsigned long Time_1=500;
const unsigned long Time_2=800;
const unsigned long Time_3=1500;
const unsigned long Time_4=5000;
char cnt=0;
char state=0;
char flag=0;
void interrupt()
{
char tmp;

  if (INTCON.F1==1)
  {
  INTCON.F1=0;  // clears  (external interrupt)
  }
//---------
 if(PIR1.RCIF==1)// If there is something in RX buffer
{
PIR1.RCIF=0 ; //Clear flag
tmp=RCREG ;
   switch(state)
   {
    case 0:{if(tmp=='*')state=1;break;}
    case 1:{
         if(tmp=='A')
         {

PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b01001010;
PORTD = 0;
delay_ms(Time_2);
PORTB = 0b01001100;
PORTD = 0;
delay_ms(Time_4);
         }
    else if(tmp=='B')
    {
PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b01010001;
PORTD = 0;
delay_ms(Time_2);
PORTB = 0b01100001;
PORTD = 0;
delay_ms(Time_4);
    }
    else if(tmp=='C')
    {

PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b10001001;
PORTD = 0;
delay_ms(Time_2);
```

```
PORTB = 0b00001001;
PORTD = 1;
delay_ms(Time_4);


  }
  state=0;break;}
 }


   }
   }

void main() {
OSCCON=0x72;

PIE1.RCIE   = 1;
INTCON.PEIE = 1;
INTCON.GIE  = 1;

TRISB=0;
TRISC=0;
TRISD=0;
PORTB=0;
PORTD=0;
PORTC=0;
UART1_Init(9600);
while(1)
{

PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b01001010;
PORTD = 0;
delay_ms(Time_2);
PORTB = 0b01001100;
PORTD = 0;
delay_ms(Time_3);

PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b01010001;
PORTD = 0;
delay_ms(Time_2);
PORTB = 0b01100001;
PORTD = 0;
delay_ms(Time_3);

PORTB = 0b01001001;
PORTD = 0;
delay_ms(Time_1);
PORTB = 0b10001001;
PORTD = 0;
delay_ms(Time_2);
PORTB = 0b00001001;
PORTD = 1;
delay_ms(Time_3);



} //close while loop
} //close void main
```

SWETCH PIC C...

```
void main() {
OSCCON=0x72;
TRISA=1;
PORTA=0;
PORTB=0;
TRISB=1;

UART1_Init(9600);
while(1)

{

if(button(&PORTB,0,50,0))

{
UART1_Write_Text("Emergency");
delay_ms(5000);
}

} //close while loop
} //close void main
```

SWETCH PIC CODE :

```
void main() {
OSCCON=0x72;
TRISA=1;
PORTA=0;
PORTB=0;
TRISB=1;

UART1_Init(9600);
while(1)
{

if(button(&PORTB,0,50,0))
{
UART1_Write_Text("Emergency");
delay_ms(5000);
}

} //close while loop
} //close void main
```